

SPACECRAFT FLIGHT SIMULATION:
A HUMAN FACTORS INVESTIGATION INTO THE
MAN-MACHINE INTERFACE BETWEEN AN ASTRONAUT
AND A SPACECRAFT PERFORMING DOCKING MANEUVERS
AND OTHER PROXIMITY OPERATIONS

Adam R. Brody

CONTRACT ^{NAGW-21}~~NAS2-11555~~
September 1988



(NASA-CR-177502) SPACECRAFT FLIGHT
SIMULATION: A HUMAN FACTORS INVESTIGATION
INTO THE MAN-MACHINE INTERFACE BETWEEN AN
ASTRONAUT AND A SPACECRAFT PERFORMING
DOCKING MANEUVERS AND OTHER (Massachusetts

N89-29020

Unclass
G3/53 0231748

SPACECRAFT FLIGHT SIMULATION:
A HUMAN FACTORS INVESTIGATION INTO THE
MAN-MACHINE INTERFACE BETWEEN AN ASTRONAUT
AND A SPACECRAFT PERFORMING DOCKING MANEUVERS
AND OTHER PROXIMITY OPERATIONS

Adam R. Brody

~~Sterling Federal Systems, Inc.~~ *Massachusetts Institute of Technology*
~~1121 San Antonio Road~~ *Space systems Lab*
~~Palo Alto, CA 94303-4380~~ *Cambridge, MA 02139*

Prepared for
Ames Research Center
Under Contract ~~NAS2-11555~~ *NAGW-21*



National Aeronautics and
Space Administration

ERRATA

NASA Contractor Report 177502

Spacecraft Flight Simulation: A Human Factors Investigation Into the Man-Machine Interface Between an Astronaut and a Spacecraft Performing Docking Maneuvers and Other Proximity Operations

Adam R. Brody
September 1988

Incorrect information was listed on the cover, title page, and report documentation page of the subject report. Please make pen and ink changes to reflect the following:

1. On the cover, title page, and report documentation page (block 11), the contract number should be changed to NAGW-21.
2. On the title page and the report documentation page (block 9), the contractor name and address should be listed as

Massachusetts Institute of Technology
Space Systems Lab
Cambridge, MA 02139

Issued 10/31/88

Table of Contents

List of Figures	5
1. Introduction.....	7
2. Background.....	9
2.1. History.....	9
2.1.1. Gemini.....	9
2.1.2. Apollo	9
2.1.3. Space Shuttle/Space Station.....	10
2.1.4. Automatic Rendezvous and Docking	10
2.2. Simulation and Design Theory.....	11
2.2.1. Chaser Craft Hardware.....	11
2.2.2. Chaser Craft Software.....	12
2.2.3. Target Craft Hardware.....	12
2.2.4. Target Craft Software.....	13
2.2.5. Mission Models	13
2.2.5.1. Rendezvous	13
2.2.5.2. Docking.....	13
2.2.5.3. Other	14
3. Design.....	15
3.1. Hardware.....	15
3.1.1. General Layout.....	15
3.1.2. Controls.....	16
3.1.3. Displays.....	19
3.1.4. Zero-g Chair.....	19
3.2. Software	23
3.2.1. Input.....	24
3.2.2. Equations of Motion	24
3.2.3. Graphics.....	25
3.2.3.1. Depth Perception.....	28
3.2.3.2. Scaling.....	29
3.3. Experimental Design	30
3.3.1. The Problem	30
3.3.2. Space Station Operational Control Zones	30
3.3.3. The Mission.....	31
4. Experimental Set-Up and Procedure.....	33
5. Simulator Evaluation.....	36
6. Experimental Results	37
7. Discussion--Analysis	49
8. Conclusions.....	51
9. Recommendations for Future Work	52

Appendixes	
A.1. Computer Code.....	53
A.2. Transformation Matrices.....	81
A.3. NASA Approach Profile.....	82
A.4. Fuel Consumption Calculation.....	83
A.5. Test Subject Training Manual.....	84
A.6. Raw Data.....	89
References Cited.....	95

List of Figures

3.1.	Cockpit (top view)	16
3.2.	Digital Auto Pilot Panel.....	18
3.3.	Center Panel.....	20
3.4.	Zero-g Posture.....	21
3.5.	Cockpit (side view) and Zero-g Chair	22
3.6.	Simulator Flow Chart.....	23
3.7.	Space Station Model	27
4.1.	Docking Target and Docking Target (detail)	35
6.1.	Mission Duration Averages	37
6.2.	Fuel Consumption Averages.....	38
6.3.	Total Cost Averages.....	39
6.4.	Theoretical Minimum Fuel and Time.....	40
6.5.	"Convenience Time" Averages	41
6.6.	Radial Fuel Averages	42
6.7.	Unsuccessful Dockings (by subject).....	43
6.8.	Sum of Mission Duration Averages vs. Unsuccessful Attempts	44
6.9.	Sum of Fuel Consumption Averages vs. Unsuccessful Attempts	45
6.10.	Mission Duration Averages for 6 Safest Subjects	46
6.11.	Fuel Consumption Averages 6 Safest Subjects.....	47
6.12.	Unsuccessful Attempts for 6 Safest Subjects	48
A.5.1.	Cockpit (top view)	86
A.5.2.	Digital Auto Pilot Panel.....	86
A.5.3a.	Range Rate vs. Range--NASA rate profile.....	87
A.5.3b.	Range Rate vs. Range--new rate profile.....	87
A.5.4a.	Docking Target	88
A.5.4b.	Docking Target (detail)	88
A.6.1.	Raw Data--Test Subject 1	89
A.6.2.	Raw Data--Test Subject 2	89
A.6.3.	Raw Data--Test Subject 3	90
A.6.4.	Raw Data--Test Subject 4	91
A.6.5.	Raw Data--Test Subject 5	91
A.6.6.	Raw Data--Test Subject 6	92
A.6.7.	Raw Data--Test Subject 7	92
A.6.8.	Raw Data--Test Subject 8	93

A.6.9.	Raw Data--Test Subject 9	93
A.6.10.	Raw Data--Test Subject 10.....	94

1. Introduction

We shall be better and braver and less helpless if we think that we ought to enquire than we should have been if we indulged in the idle fancy that there was no knowing and no use in seeking to know what we do not know.

Plato, circa 400 BC

When it becomes necessary to develop and construct a space station in earth orbit, information concerning pilot performance in docking maneuvers and other proximity operations becomes essential. Since time, fuel and other resources are at a premium in space, these parameters should be optimized to minimize mission cost. One invaluable tool both for acquiring typical mission data and for providing crew training is a computer-based flight simulator.

The support and maintenance operations of a space station will require a fleet of small spacecraft for such missions as damage patrol, repair, and satellite retrieval. Each craft must have the capability for guidance and control and must be able to rendezvous and dock at the end of the mission. A high degree of autonomy would be preferable to alleviate demands on ground control or on the space station itself.

The design, development, testing and evaluation of a spacecraft flight simulator actively incorporate many different disciplines. For convenience, these may be broken down into two broad categories: hardware and software. Hardware refers to control station (cockpit) design with a working understanding of anthropometrics and ergonomics. Software includes both the interactive graphics algorithms which serve as interfaces between controls and displays, and the continuous updating and maintenance of the values of certain critical vehicle parameters such as altitude, attitude, range and operational status.

One of the most important preliminary decisions to be made in designing a simulator is the determination of the range of missions which the vehicle to be simulated will perform. While there is some important history and evolution of rendezvous, docking and other proximity operations (proxops) techniques, the frequency with which satellites will be retrieved and the requirements for operating a permanently manned space station in Low Earth Orbit (LEO) demand that existing techniques and procedures be reevaluated and new ones be created. While the selection of lunar orbit rendezvous mode in the Apollo program defined the initial research in rendezvous, docking and proxops, the emergence of new ranging and control technologies, the increase in traffic, and the differences in missions and vehicles call for further investigation.

According to the Mission Operations Directorate at the Johnson Space Center, spacecraft rendezvous refers to "all orbiter or payload maneuvers (orbit shaping, phasing intercept initiation) up to initiation of proximity operations." It is usually performed in preparation for other activities such as docking. Docking is a natural successor to rendezvous and enables crew transfer and vehicle or space station resupply. Proximity operations occur during the "post rendezvous phase where the relative separation range and range rate are sufficiently small (<1000 feet, <1 foot per second [300m, .3m/s]) that rendezvous operations are not required to restore proximity". A small set of proxops includes stationkeeping, approaches, departures, inspections and rescues. [NASA Johnson, 1985]

The purpose of this investigation is to examine and evaluate the role of manual control in the aforementioned missions. Manual control of spacecraft with direct or remote visual cues is conceptually the simplest control method, and in most space missions to date has been the primary or back-up control scheme. Manual control eliminates some of the need for elaborate computer control algorithms and sophisticated image processing equipment. Where weight or power restrictions prohibit the utilization of transponders on the target craft, it is the only available method.

2. Background

No man can reveal to you aught but that which already lies half asleep in the dawning of your knowledge.... The astronomer may speak to you of his understanding of space, but he cannot give you his understanding..... For the vision of one man lends not its wings to another man.
Kahlil Gibran, 1923

Manual control may take place either on-site or remotely. In either case, the current state of the art requires heavy support with ground operations. For the reasons mentioned earlier, manual control is fundamentally simpler than automatic docking.

2.1 History

2.1.1 Gemini

Investigations of rendezvous and docking procedures arose in the evolution of the U.S. space program once the initial exploratory phase (Mercury) had been successfully completed and missions became more ambitious. In the United States, the Gemini program was used to acquire these techniques and develop these technologies, and to give astronauts the practice they needed to get to the moon. Orbital rendezvous procedures were performed as the various Gemini craft tracked and approached their respective rendezvous targets. Gemini demonstrated that "precise flight-crew responses during orbital flight is [sic] critically dependent upon the fidelity of the simulation training received prior to flight." [NASA Office of Technology Utilization, 1967]

2.1.2 Apollo

The Apollo program made extensive use of experience acquired from Gemini. Two rendezvous and docking operations were necessary to reach the moon. First, after reorientation, the command/service module (CSM) docked with the lunar module (LM). Later, after rising from the lunar surface, the ascent stage docked with the CSM. A brief examination of the man-machine interface at these two crucial points in the mission is worth discussing here.

CSM Rendezvous Navigation was accomplished by a combination of visual sensing and manual control which gave alignment, after which the state vectors were used to compute maneuvers. Updates were received automatically by a VHF radio link for ranging, and optically by an astronaut looking through the crew optical alignment sight (COAS). In this procedure, the astronaut sighted a high intensity strobe tracking light and initiated a mark on the computer. The computer recorded the time and optics shaft, trunnion and inertial platform gimbal angles from which it updated the LM state vector.

The LM Guidance Computer (LGC) calculated maneuvers from range, range rate and direction data obtained from the LM rendezvous radar (RR). Tracking the CSM transponder occupied the LM commander full time as he provided the computer with an update each minute.

The abort guidance computer (AGC) occupied the LM pilot (LMP) full time as he manually keyed navigation marks, range and range rate data every two minutes. From these data, the AGC computed updates. Unlike the CSM, the LM lacked a hardware interface with the rendezvous radar so shaft and trunnion angles had to be zero. [Hughes, 1970]

Rendezvous and docking were also necessary for the Apollo Soyuz Test Project (ASTP). In that case, an androgynous docking system accommodated large misalignment errors and a range of contact velocities. The Apollo CSM was designated the chaser craft because of its larger fuel capability, while the Soyuz was outfitted with a transponder, passive transmitter-receiver with automatic response, white flashing beacons and two docking targets, and was painted part white and part green to facilitate optical recognition. Use of the CSM as chaser vehicle allowed for operations similar to those used during the Apollo program and for the three CSM dockings in the Skylab program.

2.1.3 Space Shuttle/Space Station

In the space station era many craft will be in operation simultaneously. Since Mission Control at the Johnson Space Center in Houston lacks the capability for monitoring many vehicles on-orbit, a greater degree of spacecraft autonomy is necessary for the performance of concurrent missions. Creating a Mission Control to support each orbiting craft would be a tremendous effort. Rather, reliance on ground support should be reduced and the effect of this reduction on on-board control must be understood.

The shuttle has successfully approached and grappled several satellites, but has yet to dock with anything. Currently, a star tracker is used for far field sighting, and rendezvous radar is used at smaller distances (less than 300 n mi [556 km]). Optical sighting is always available as a back-up. Once the space station has been installed, a wealth of new technologies will become available to facilitate rendezvous and docking operations. Most important are those concerned with ranging. Global positioning system (GPS) receivers (or a similar ranging technology) on both vehicles will allow for more autonomous capability by providing more accurate and updated proximity information than is currently available. [NASA Johnson 1985]

Clearly the time for optimizing rendezvous and docking procedures is now; computer simulation is an excellent tool for evaluating innovations in these areas.

2.1.4 Automatic Rendezvous and Docking

It should be noted that as far back as 1967, the USSR achieved an automatic docking with Cosmos 186/188. [Novikov, 1968] The obvious question is, "Why should the United States space program devote so much effort to developing manual techniques when automation of these procedures is an established technology?" The answer is that, with a simple, coherent, well-defined mission model, automatic docking is possible and practical. However, for patrolling the surroundings of a space station in search of a damaged solar array, for approaching a spinning satellite whose exact position and orientation are unknown, or simply for attempting rendezvous with any object which cannot or will not provide the feedback for an automatic mechanism, manual (astronaut) control is required. As is the case with all other manned systems, the human operator provides flexibility with his unmatched capacities of perception, judgement, dexterity and imagination.

While it is relatively easy to have two spacecraft automatically dock when their launch times, altitudes, and range are precisely known, exact range data would not be available for a shuttle and satellite which have been in orbit for some time. Ranging technologies such as the Global Positioning System (GPS) require cooperative receivers to be implanted on the target craft. While it is likely that a space station docking port would be equipped with such a device, it may become necessary to traverse a section of the station which is not so instrumented. This may occur if a solar panel needs repair, for example. In such a case, a human pilot is required.

It should also be noted that the space shuttle has the capability for a fully automatic landing with manual control as a backup, and yet manual control has always been the only method used. This emphasizes the importance and utility of human control in the U.S. space program in the presence of an automatic control alternative.

2.2 Simulation and Design Theory

As mentioned earlier, flight simulation is a valuable tool not only as a low cost, low risk training device, but also for determining the human factors needs of the pilot in the cockpit. Simulators provide for rapid and inexpensive alterations in the layout of controls and displays. The type and variety of displays can also be changed with great ease in order to determine the optimal flight configuration as far as operator preference, performance and productivity are concerned.

Some of the proxops design issues which may (and should) be analyzed and studied with an accurate flight simulator are plume impingement, collision avoidance, stationkeeping/formation flying techniques, 3-D translation and attitude control, accommodation of tumbling/non-cooperative spacecraft, and controller algorithm design/analysis technology. [NASA Johnson, 1985] The diversity of this list further demonstrates the required capabilities of a good simulator. Those issues associated with human performance are emphasized here.

2.2.1 Chaser Craft Hardware

While software design plays an integral role in the creation of a flight simulator, good hardware design cannot be undervalued. The importance of good hardware design is indicated by the amount of human factors research devoted to it. The Teleoperator and Robotics Evaluation Facility (TOREF) at the Marshall Space Flight Center (MSFC) has been researching spacecraft cockpit design since 1971. Their studies indicate that a complex control station requires more than the usual human factors guidelines and standards in its design. Following is a discussion of some design criteria associated with control station design. [NASA Marshall, 1984]

The cockpit should accommodate the 5th to the 95th percentiles of the user population. Most control stations utilize visual feedback via imaged scenes and alphanumeric characters as their primary display mode. The operator must have complete control over contrast, brightness, focus, display stability, viewing angles and ambient illumination. Monitor size is also important: while larger screens may provide a higher degree of resolution with big images, smaller screens demand less head and eye movement (which lead to fatigue) and require less panel space. To minimize cosine error, the screens should also be perpendicular to the pilot's line of sight.

Hand controller operations should minimize fatigue by means of full arm supports and negative attitude work surfaces. Six degrees of freedom (DOF) are controlled with two 3 DOF hand controllers. The right hand is delegated rotation controller because of its capability for greater precision while the left hand can manage translation maneuvers in an acceleration, On/Off control mode.

Continuous joystick controls may be either linear or non-linear. While linear controls provide uniform responses and, in general, are more predictable, non-linear controls permit small, precise adjustments as well as large, coarse movements. In either case, the system resolution limits the accuracy. Also, the effort required for control operation must be great enough to minimize accidents yet low enough to minimize fatigue. Giving the pilot the ability to interactively choose his control modes should help optimize operations.

2.2.2 Chaser Craft Software

Software specifications for the chaser vehicle which is being simulated here center on the flight control system. For controlling translation, open-loop on/off acceleration mode provides for good fuel economy and responsive handling. Pulse mode allows precision adjustments. Closed-loop position control is not practical because of its reliance on elaborate position-sensing equipment.

To control rotation, open-loop acceleration mode may be difficult because of gravity gradient disturbances. Closed-loop rate-command mode requires rate gyros and a phase-plane autopilot (or similar devices), but reduces pilot workload and allows for automatic disturbance compensation and inertially stabilized or Local Vertical-Local Horizontal (LVLH) flight. An attitude rate-hold feature is a simple and inexpensive method for matching target tumbling rates and for reducing pilot workload.

2.2.3 Target Craft Hardware

An important item in design requirements/specifications is the space station docking module. The most recent analysis (from the proceedings of a 1985 NASA workshop on rendezvous and proxops) specifies axial velocity to be between 0.05 and 0.15 meters/second, a lateral velocity of 0.06 m/s and an angular velocity of 0.6 deg/s. The lateral misalignment should be no more than 0.23 m and angular misalignment should not exceed 5.0 degrees in the roll and 6.0 degrees in the pitch and yaw planes. [NASA Johnson, 1985] These specifications are only slightly more stringent than those prescribed by the International Spacecraft Docking Agreement of 1973. It should be understood that the smaller the allowable capture radius and alignment angle ranges are made, the greater the operational cost in fuel and time.

The target vehicle should be cooperative to the greatest extent possible. This involves the installation of reflective devices or receiver/transmitters to enable range determination. Reflectors have the advantage of being passive (requiring no power and having fewer parts to wear out or break) as well as providing a means for attitude measurements. Electronic ranging techniques, on the other hand, may operate over greater ranges and larger angular misalignments.

2.2.4 Target Craft Software

The particular target used in this study, a cooperative space station, is capable of assisting docking and rendezvous maneuvers by providing information necessary for ranging calculations. A combination of reflectors and passive transmitter/receivers can provide a range accuracy on the order of 1 cm and velocity accuracy on the order of 1 cm/s.

Target craft software automatically awaits an interrogation signal from the chaser and responds with a message from which range and range rate calculations may be made. These values will be correct only for their signal source, the docking port. From these values and a software model of the station, distances and approach velocities to other locales of the target craft may then be calculated.

2.2.5 Mission Models

As mentioned earlier, there are many mission models worth analyzing. These may be broken down into three categories: rendezvous, docking, and "other". These will be briefly described here.

2.2.5.1 Rendezvous

Rendezvous phasing orbits are those which establish the chaser and target in coaxial orbits. After achieving a common line of apsides and adjusting the phasing and altitude, the objective is to put the chaser vehicle in a lower coelliptic orbit. Additional maneuvers include: a terminal interception maneuver, midcourse and braking maneuvers to establish velocities, and stationkeeping.

A "Low-Z" approach maneuver may be appropriate when plume impingement poses a problem. On the shuttle, this maneuver consumes propellant twelve times faster than otherwise while providing only one ninth the thrust and is therefore reserved for the final 60-90 m.

A direct approach method has advantages in timing and propellant consumption but may contribute dangerous thruster plume effects during the braking maneuver.

Approaches along the radius vector are known as "R-bar". In this method, the target is approached from below and orbital mechanics effects provide a natural braking mechanism. This helps to alleviate plume impingement effects. Rendezvous along the velocity vector is known as "V-bar".

Other methods exist, such as "one impulse" and " Δh drift", which are not relevant to this discussion.

2.2.5.2 Docking

Four mission models for docking can be detailed according to target spacecraft dynamics. Inertially stabilized approach entails flying straight toward the target and matching attitude while approaching the docking fixture.

Single axis roll rate also involves flying straight toward the target while matching roll rate and angle. Pitch and yaw angles are corrected during the final approach.

The single axis pitch or yaw rate method consists of approaching the target on a trajectory lying in the plane of rotation. The docking fixture is intercepted as it becomes aligned with the chaser spacecraft. Forward and lateral thrust may be required to correct for timing errors. Attitude rate hold mode is used to maintain angular alignment during the final phase.

Multiple axis attitude rates refers to the combination of the above techniques when target motion prediction is more difficult. Small translational and angular adjustments are almost always required in this situation.

2.2.5.3 Other

Other missions an astronaut in an orbiting pod-type craft may have to perform are stationkeeping, damage patrol, construction/assembly, servicing/resupply, retrieval, rescue and repair. These should be extensively simulated as well to learn how to best perform them on orbit.

3. Design

*To be able to rise from the earth;
to be able, from a station in outer space,
to see the relationship of the planet earth to other planets;
to be able to contemplate the billions of factors
in precise and beautiful combination that make human
existence possible;
to be able to dwell on an encounter of the human brain and
spirit with the universe--
all this enlarges the human horizon....
Norman Cousins, 1973*

3.1 Hardware

To maximize productivity and performance, it is important when designing a cockpit to avoid the characteristic disadvantages of a conventional work station. Among these are fatigue, incompatible eye/hand feedback, excessive head and hand movements, and unnecessarily increased operations time and error rates. The three main pieces of hardware in the cockpit are the controls, the displays and the zero-g chair.

3.1.1 General Layout

Anthropometrics is the study and cataloging of human body proportions and dimensions. The discipline of human factors engineering, which makes extensive use of anthropometrics, became more systematic during World War II as airplanes were being designed for military use. [McCormick, 1982] Every effort was made to accommodate as much of the user population as possible by placing buttons, switches and other controls within the pilot's reach envelope and by installing displays along optimal sight lines. A large body of accumulated data detailing every imaginable body dimension, is cross-referenced by percentile and age group for ease in identifying these dimensions.

A special chair was developed for these tests which assists the test subjects in maintaining a position similar to the zero-g neutral body posture. Since this zero-g chair is new and unconventional, many body parameters such as "popliteal height" and "sitting height" must be subjected to a modified interpretation or simply ignored. The two most important values are "thumb tip reach" (26.7 inches) and viewing distance (18-22 inches). The thumb tip reach length used here was a projection of the fifth percentile length for a female military population in 1985. (The male length and those for the ninety-fifth percentile for women are greater. The reach must be small enough to accommodate users from the lower end of the spectrum.) [Panero, 1979] Figure 3.1 shows a top view of the cockpit. Some user-adaptability is available by moving the zero-g chair.

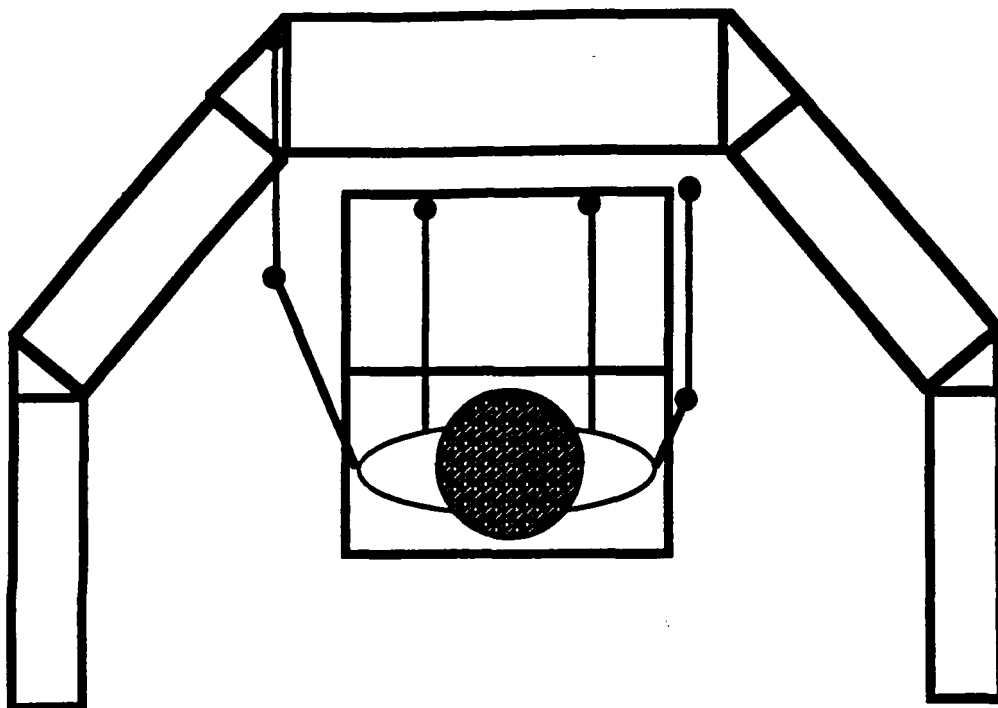


Figure 3.1: Cockpit (top view)

3.1.2 Controls

A glance inside the cockpit of the Space Shuttle reveals a plethora of buttons and switches to control the vast complex of machinery that is the vehicle. Even removing those items associated with launch and landing (which are not of concern here) leaves an assortment that would take the experienced pilot months (at least) to learn. Under the conditions guiding this design, it is grossly impractical to present subjects with an experimental device requiring extensive training. For this reason, and to simplify the experiment by reducing the number of variables, the operator activity is focused on the flight control system. Environmental regulation and other such controls are omitted. The flight control system is composed of two 3 degree of freedom (DOF) hand controllers and twenty buttons serving as an interface to the vehicle's digital auto pilot (DAP).

In the evolution of spacecraft it is important to maintain a high degree of uniformity in control layout and operation to facilitate learning and performance as well as research and production. In the same way that an experienced motor vehicle operator can drive a previously unseen car after a minimal familiarization period, current shuttle pilots should not require extensive retraining to operate the next generation of spacecraft.

The hand controllers described here are virtually identical to those currently used on the shuttle. The left one is used for translation commands and is mounted horizontally. Movements of the stick are directly analogous to movements of the spacecraft in each of the three axes.

The right hand controller is used for rotational control. Pitch and roll commands are initiated the way they would be in an aircraft. Yaw commands are sent by twisting the stick in the appropriate direction. Both sticks were manufactured by Measurement Systems Incorporated (model numbers 544 and 544-G510).

The DAP (Figure 3.2) is a collection of twenty buttons which are used to individually select the control mode under which each DOF is operating, three for each DOF. Also included are an attitude hold button and an enable button. In rotation, this DAP is slightly different from shuttle's DAP in that hand controller deflection not only indicates direction but also specifies rate. The DAP is located adjacent to the translational hand controller for convenience.

DIGITAL AUTO PILOT

TRANSLATION			ROTATION		
X	Y	Z	ROLL	PITCH	YAW
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
LOW	LOW	LOW	RATE	RATE	RATE
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
HIGH	HIGH	HIGH	ACCEL	ACCEL	ACCEL
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PULSE	PULSE	PULSE	PULSE	PULSE	PULSE

DAP	ATT
<input type="checkbox"/>	<input type="checkbox"/>
ON	HOLD

Figure 3.2: Digital Auto Pilot Panel

3.1.3 Displays

In the same way that there are optimal positions for controls to minimize fatigue while maximizing performance, one can and should arrange and orient displays to effect similar benefits. Although the optimal primary sight line for a zero-g environment falls fifteen degrees below the level for a one-g environment, the cockpit is designed to optimize earth-bound operation and the displays are designed accordingly.

The large twenty-five inch color monitor in the center panel represents the forward-facing window of the spacecraft. This screen simulates what an operator would actually see on orbit. The smaller, eight inch monitors display such critical vehicle parameters as altitude, attitude, range, range rate and system status. NASA concluded after Gemini that the crew could accomplish all rendezvous maneuvers with these data. [NASA Office of Technology Utilization, 1967] Figure 3.3 shows the astronaut's view of the center panel.

3.1.4 Zero-g Chair

Figure 3.4 depicts the posture that the human body assumes in the absence of a gravity field. While close to a standing position, the bends in the legs indicate a more comfortable, relaxed position, possible only when the body's weight need not be supported. A "zero-g chair" (Figure 3.5) is utilized to aid the body in maintaining this position while stationed in the simulator.

Fabricated out of wood with neoprene cushioning, the chair is a compromise between a comfortable one-g position and a free-floating zero-g stance.

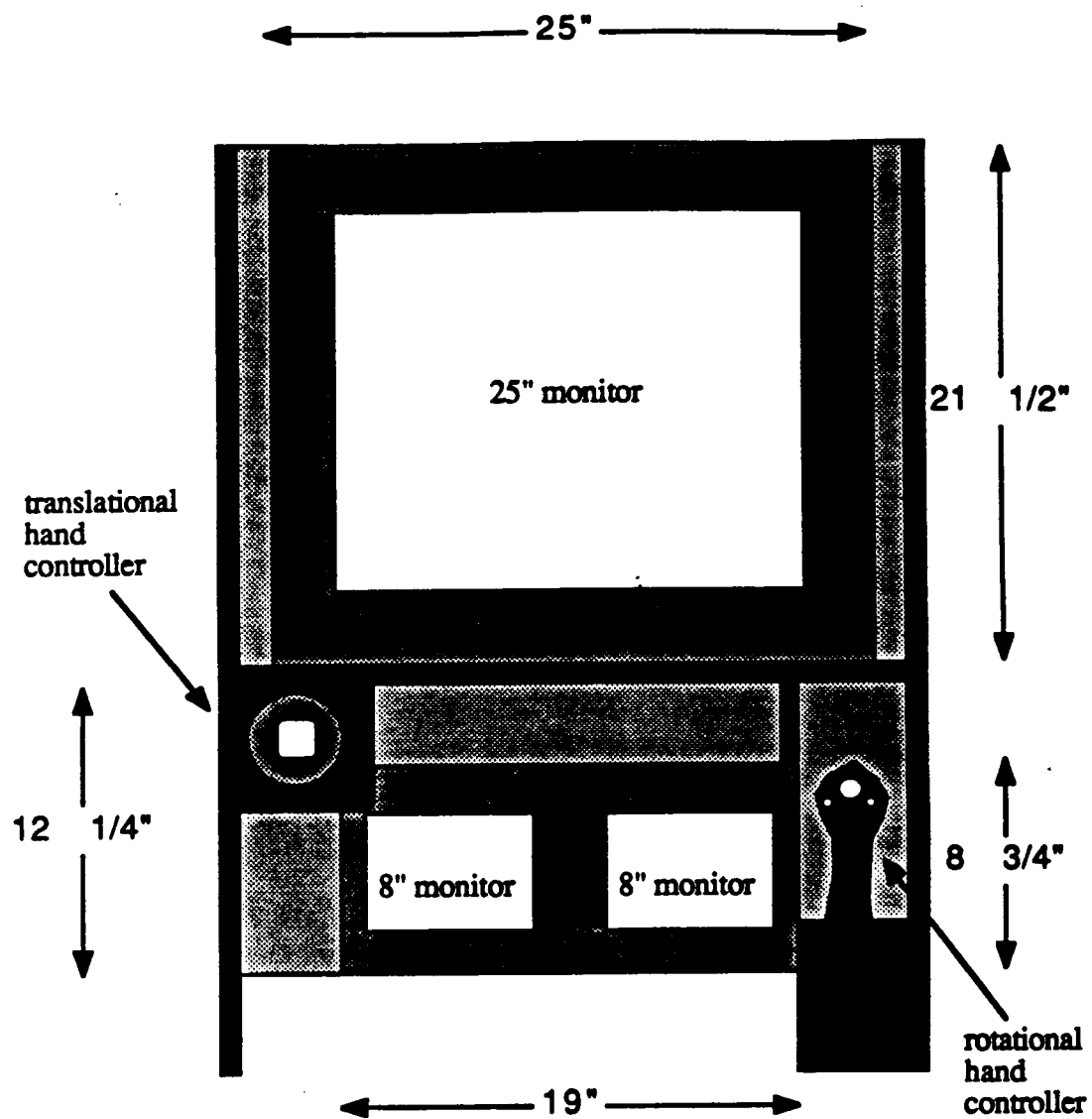


Figure 3.3: Center Panel

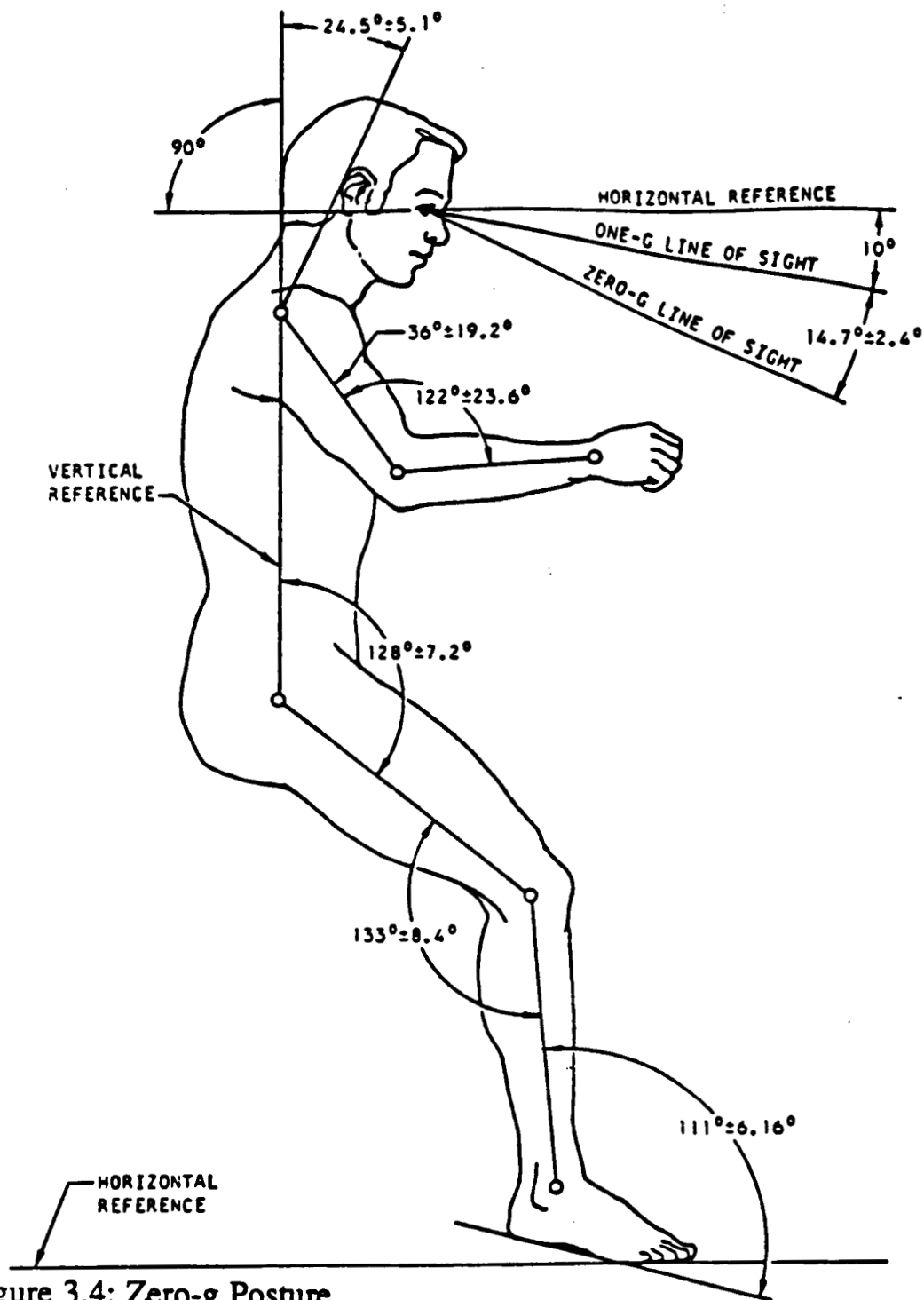


Figure 3.4: Zero-g Posture

(From MSFC-STD-512A, 1976)

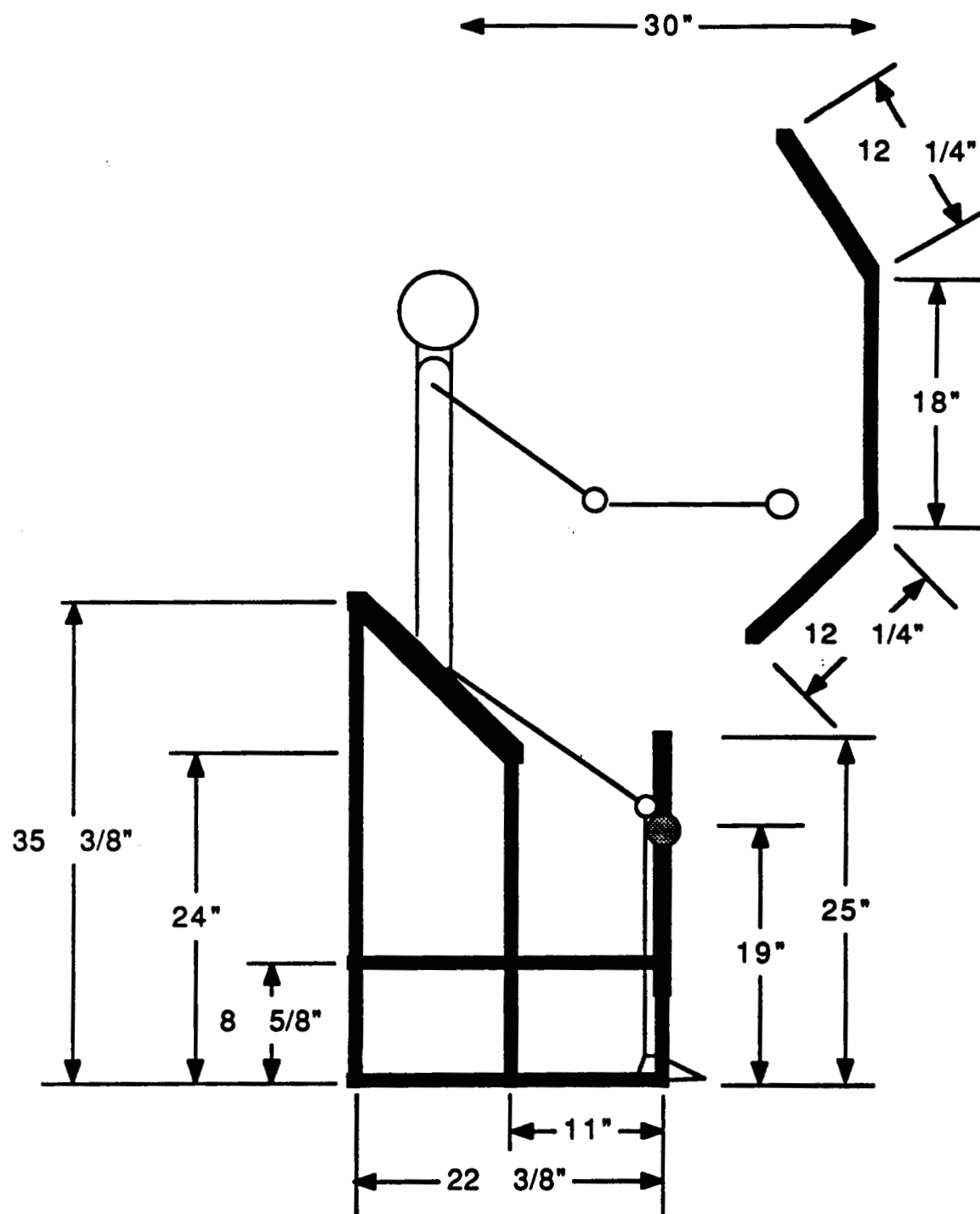


Figure 3.5: Cockpit (side view) and Zero-g Chair

3.2 Software

Two computers are required to operate the flight simulator. An IBM Personal Computer performs the A/D conversions of hand controller voltages to numbers that the rest of the software can comprehend. A Silicon Graphics IRIS 3020 Workstation performs the orbital mechanics calculations and drives the graphics display. (See Figure 3.6.) Equations of motion determine the relative position of target and chaser as a function of time.

The IBM is programmed in Advanced Basic which supports the asynchronous communications line to the IRIS. After receiving the A/D data, the IBM converts these values into update rates for the six degrees of freedom. These rate changes are transmitted to the IRIS, at a rate of 9600 baud, for processing through Hill's equations (see 3.2.2) and then to the graphics algorithms for final processing. The IRIS computer is programmed in C.

(See Appendix A.1 for all computer code.)

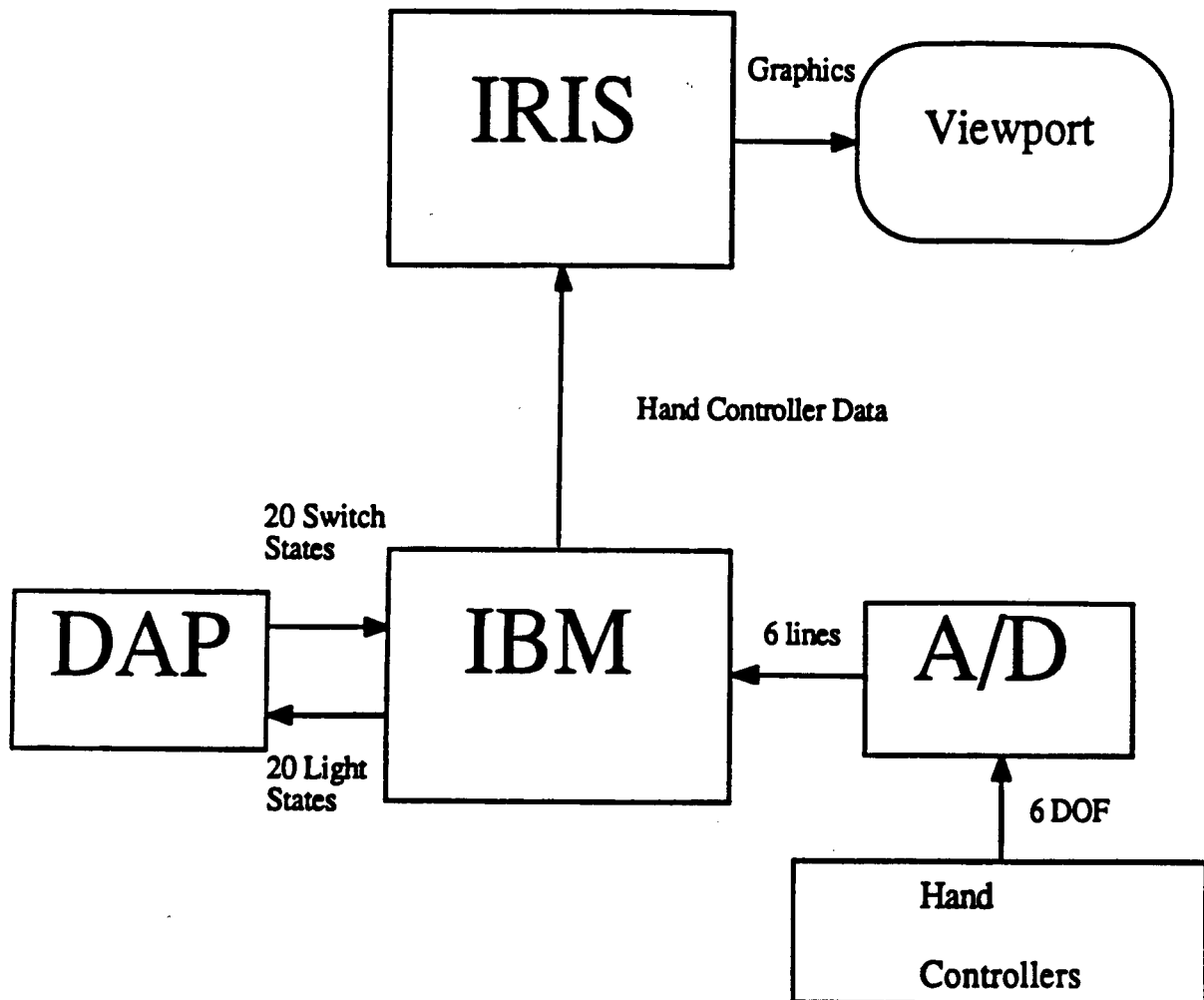


Figure 3.6: Simulator Flow Chart

3.2.1 Input

Each movement of the hand controllers has three modes of interpretation, depending upon which of three buttons is activated. To send a command to the thrusters, a pilot must choose one of three control modes for the particular DOF to be operated. Translation modes are all open-loop, acceleration control; the pilot selects High, Low, or Pulse operation. In High and Low modes, the direction of acceleration is indicated by the deflection of the controller and is maintained at a constant rate for the duration of the command. Pulse mode provides a single minimum impulse regardless of how long or how far the stick is deflected.

For the rotation controller, rate, acceleration and pulse control modes are available. Rate provides a rotation rate proportional to controller deflection: there are eleven discrete intervals--five plus, five minus, and a dead zone in the middle. Acceleration and Pulse modes provide responses similar to their translational counterparts.

3.2.2 Equations of Motion

A spacecraft in orbit around the earth obeys laws which fundamentally serve to balance kinetic energy and the gravitational potential. The most general form is the vis-viva equation, namely,

$$v^2 = \mu_E \left(\frac{2}{r} - \frac{1}{a} \right)$$

where v is the orbital velocity, μ_E is the gravitational constant, equal to $398604 \text{ km}^3/\text{s}^2$ for the earth, and a is the semi-major axis of the elliptical orbit. For a circular orbit, a is the radius and

$$v = \sqrt{\frac{\mu}{r}}$$

The orbital period is

$$P = 2\pi \sqrt{\frac{a^3}{\mu}}$$

The equations of motion which govern the relative motion between one body in a uniform circular orbit (space station) and another body (orbiting spacecraft) are known collectively as the Clohessy-Wiltshire solutions to Hill's equations. These solutions describe relative position as a function of time and are called by the simulator software whenever a disturbing impulse (thrust) is introduced. The closed forms are:

$$X = \frac{V_{x0}}{n} \sin(nt) - \left(2\frac{V_{y0}}{n} + 3X_0 \right) \cos(nt) + 2\frac{V_{y0}}{n} + 4X_0$$

$$Y = 2\frac{V_{x0}}{n} \cos(nt) + \left(4\frac{V_{y0}}{n} + 6X_0 \right) \sin(nt) + (Y_0 - 2\frac{V_{x0}}{n}) - (3V_{y0} + 6nX_0)t$$

$$Z = Z_0 \cos(nt) + \frac{V_{z0}}{n} \sin(nt)$$

where X is measured radially outward, Y is along the velocity vector and Z is positive out of the orbital plane to the left. The mean orbital motion, n, is equal to:

$$\sqrt{\frac{\mu}{a^3}}$$

For a 300 km (162 n mi) orbit around the earth, the period is 90.5 minutes, $n=0.001158$ rad/s and the circular orbital velocity is 7.73 km/s. [Kaplan, 1976]

Time for Hill's equation's is measured from the incidence of thrust and all values are initialized after each thrust. The time derivative of each of these equations yields the corresponding velocity equations:

$$V_x = V_{x0} \cos(nt) + (2V_{y0} + 3nx_0) \sin(nt)$$

$$V_y = -2V_{x0} \sin(nt) + (4V_{y0} + 6nx_0) \cos(nt) - (3V_{y0} + 6nx_0)$$

$$V_z = -Z_0 n \sin(nt) + V_{z0} \cos(nt)$$

3.2.3 Graphics

A space station model is "drawn" in the memory of the IRIS computer. (See Figure 3.7.) By keeping track of the location of the chaser craft, the computer performs the matrix transformations necessary to produce the correct image on the screen. The image is shown in perspective with appropriate clipping on the near and far planes.

A system known as double buffering controls the refresh rate of the screen. The contents of one buffer are presented while the other buffer is being updated and then they are switched.

A transformation matrix determines how the space station appears on the screen. A perspective matrix, whose arguments are field of view in the y direction, aspect ratio, and distance to the near and far clipping planes, is initially loaded on the matrix stack. It is subsequently multiplied by one rotation matrix for each axis and a translation matrix for all three axes. The station is then called and presented in the correct size, orientation, and location. (See Appendix A.2.)

After the station is drawn on the screen, it is redefined to include the current transformation matrix. This is done recursively each time the buffers are swapped. In this way, the incremental change matrix is the only multiplier in each loop.

A scale model of the Earth is included for added realism. Since the IRIS cannot draw 3-dimensional spheres, only 2-dimensional circles, three circles intersect along a centerline at 120° intervals. The model is a uniform blue-green color simulating the Earth's appearance from orbit.

Continents, mountains and other natural features are not installed to reduce the computational and drawing time of each image. However, longitudinal lines are drawn vertically and horizontally to provide motion parallax and to aid in tracking. The Earth's image gives the pilot a better sense of his orientation in orbit. The distance from the Earth to the station, altitude, can be readily altered to produce different station altitudes. This affects the orbital period and other orbital mechanics effects.

All range and orientation information is obtained directly from the total transformation matrix, excluding the perspective matrix. This guarantees the accuracy of the data. Control stick inputs are integrated to determine range rates and rotation rates .

The calculations necessary to draw an image on the screen take a lot of processor time. In order to decrease the time spent on calculations and increase the refresh rate, a graphical procedure known as pruning is employed. A 2-dimensional boundary box is placed around each object (e.g. earth, docking port, keel truss). If the object will not appear on the screen, the calculations necessary to produce its image will not be performed and processing will shift to the next object. This was found to more than double the refresh rate and greatly enhance the smoothness of the real time simulation.

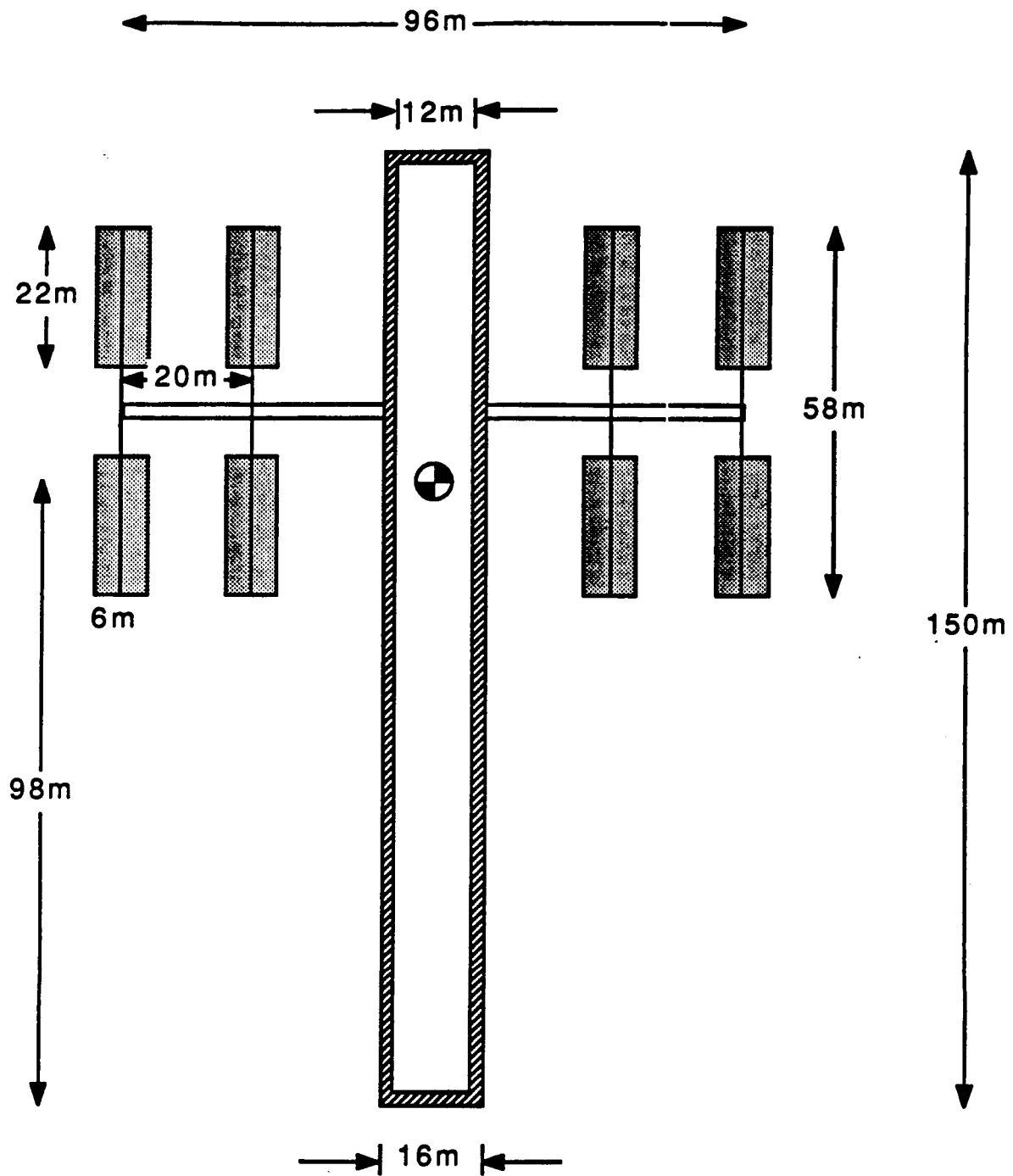


Figure 3.7: Space Station Model

3.2.3.1 Depth Perception

One potential difficulty inherent in a simulator of this type is the accurate representation of depth (distance) in a device, namely a video monitor, which is directly capable of presenting only two dimensions. Since in reality every object on the screen is the same distance from the pilot, there are no oculomotor depth cues arising from accommodation and convergence of the eyes. In other words, the amount the eyes must accommodate (focus) or converge is constant for every object on the screen. In a sense, all objects are prefocused at the distance to the screen.

Binocular visual information arising from stereopsis or retinal disparity is also missing. Since the eyes view the world from two distinct vantage points, slightly different images appear on the two retinas. The brain uses these disparities as one cue to judge distance. Again, since all objects on the screen are at the same effective distance and both eyes are converging on the same point, the retinal images from both eyes are virtually identical for objects in both fields of view and the ability to perceive depth from retinal disparity is absent.

One monocular cue, motion parallax, is also absent from two dimensional images. Moving the head with respect to a scene ordinarily provides depth information by virtue of parallax. This clue, similar to the data provided by retinal disparity, is absent from the simulator for the same reason.

Although the brain is successfully deprived of these three means of depth perception, three static depth cues remain unimpaired in the simulator. These cues depend upon geometry and illumination and were initially discovered by artists who by the fifteenth century knew how to "trick" the brain into perceiving three dimensions from two. These cues are interposition, size and perspective.

Interposition refers to the occlusion of one object by another. The brain interprets this as the second object being closer than the first. While this is a very strong cue, its operation is essentially binary. One object is either nearer or farther; information concerning amount of relative distance is unavailable.

Since the image of an object on the retina grows as the object gets closer, size is an important indication of depth. As the size of the image on the screen increases or decreases, relative distance (and some sense of velocity) may be inferred.

The third kind of depth cue, perspective, refers to geometrical variations in the appearance of an object due to differences in viewing location and angle. One type of perspective, linear perspective, was developed by Leonardo da Vinci in the fifteenth century. Linear perspective results from the two dimensional projection of a three dimensional image and is created by having the object's receding lines converge to a point.

Texture gradients are another type of perspective which provide depth cues. Researchers have shown that texture gradients provide precise and unambiguous information concerning range and attitude of surfaces and also about the sizes of objects on these surfaces.

A third type of perspective, aerial perspective, refers to the degradation in the images of distant objects caused by light passing through a greater distance in the atmosphere than with nearer objects. [Sekuler, 1985] Atmosphere, and therefore aerial perspective, does not play a role in this simulator.

In accordance with these observations, a perspective command is used to transform an orthographic image into one whose lines converge in the distance to provide depth information. While depth cueing can also be provided by having closer sections of the space station appear brighter than more distant ones, it was found that this routine took an unacceptable amount of time to operate in a real time simulator. Although atmosphere does not reduce the amount of received light, the inverse square decrease in intensity of point-source light still applies, and less light arrives from distant objects than from those that are closer to the observer. In actuality, this difference in intensity is usually imperceptible to the human eye, so sacrificing depth shading is not a major shortcoming to the simulator.

3.2.3.2 Scaling

Image size on the screen is appropriately scaled to present objects in their correct apparent size. When the pilot is correctly positioned in the simulator, the monitor screen ("window") subtends an angle of approximately 20 degrees. The graphics software accounts for this field of view angle in presenting the image.

3.3 Experimental Design

3.3.1 The Problem

Experimental design is as much of a challenge as the design of hardware or software. For instance, since a rendezvous maneuver may take up to several hours to complete without being very demanding on the operator, inactivity may induce boredom and disinterest. Unfortunately, a simulated maneuver performed at unrealistic speeds, while more interesting and less fatiguing from the subject's point of view, bears too little resemblance to reality to yield useful data. Briefly, the problem is how to keep the subject occupied and motivated while preserving the fidelity of an actual mission.

3.3.2 Space Station Operational Control Zones

NASA is planning for the environment around the space station to be divided into nine zones to serve as guidelines for orbital operations. Zone 1 is the Proximity Operations Zone, consisting of a 1 km diameter sphere centered at the space station. A rectangular volume along the orbital path extending from 37 km behind the station to 37 km in front of the station, 37 km above, 37 km below and ± 9 km out of the orbital plane constitutes Zone 2. The space station will monitor, and be capable of controlling, all unmanned spacecraft in Zones 1 and 2. After initial deployment and separation, vehicles will enter the Departure Zone (Zone 3) which reaches to 185 km in front of the station, 37 km above and below and 9 km to each side. Zone 4, the Rendezvous Zone, appears as a mirror image of Zone 3 behind the station. The Standard Orbit Rendezvous mandates that the chaser craft perform its final closing maneuvers from an offset point located behind the station. Zones 5-9 have less direct relationships to rendezvous and proximity operations. [NASA Johnson, 1984]

NASA's intention is for chaser spacecraft to arrive at an offset point of 1000 feet from the station at the completion of a rendezvous maneuver. This point is in front of the station for manned vehicles and behind it for unmanned craft. Manned spacecraft will continue in to 100 feet. This distinction is a result of lighting requirements. Ideally, approaches will take place shortly after orbital sunrise and be completed before orbital sunset to keep the light source behind the observer. A basic ground rule for daylight approaches is that the angle between the line-of-sight (LOS) to the sun and the line of sight to the target be greater than 20° . [NASA Johnson, 1984]

3.3.3 The Mission

The costs associated with space missions are formidable. In the current proposed configuration, the space station will nominally be inhabited by six astronauts for tours of duty lasting ninety days. [NASA Johnson, 1984] Operationally, the launch costs connected with getting the crew and their supplies (food, water, oxygen, and other expendables) to the station are tremendous. In 1984 dollars, it costs \$3500 per kg of payload to get these and other items into orbit aboard the shuttle and \$439,000 per day (\$5.08/second) for each person at the station. [Stuart, 1986] Since rendezvous maneuvers can take anywhere from several hours to a few days to perform [NASA Johnson, 1984], the cost of time is very significant. Costs incurred by research, development, testing, and engineering, as well as astronaut training and ground support contribute to mission expense. For these reasons, time on orbit, especially astronauts' time on orbit, is at a premium.

As mentioned earlier, dockings were performed as part of the Gemini and Apollo missions. While time was expensive then too, there was not as big an advantage to finishing a task early then as there will be in the space station era. If through pilot skill or erroneous calculation a maneuver was completed in less time than expected, then that time could be used for rest and relaxation. Mission durations were based upon timeline estimates and the time allotted was deemed sufficient for the completion of all mission objectives. A job queue, as such, did not exist.

Space station operation will be somewhat different. If a task is finished in less time than expected, then the next one will begin early. If an astronaut can safely retrieve a satellite in half the time using a new control strategy, then twice as many satellites can be retrieved during a given period of time. From an operational standpoint, this potential increase in productivity would enhance mission success and reduce operating costs.

Current shuttle guidelines suggest a "0.1% rule" for rendezvous and docking maneuvers. (See Appendix A.3.) This dictates a closure velocity of 0.1% of the range to the target per second. [NASA Johnson, 1983] At a distance of 1000 m, this corresponds to 1 m/s. From a range of 1 km, it would take approximately 1 hour to dock. The time required for this operation, assuming only one person is involved, costs about \$18,000. The Space Station Reference Configuration declares that all manned vehicles will be monitored by the space station in final rendezvous, so these costs will actually be doubled. Fuel costs would bring the total docking cost up to about \$45,000. It can very easily take a day to get a satellite and bring it to the station for repair, and an equal amount of time to return it to its former orbit. For the aforementioned reasons, it would be greatly beneficial to reduce this time overhead.

Two other reasons illustrate the importance of determining a minimum safe time for docking. Firstly, in the event of an emergency, it may be necessary to dock the spacecraft as soon as possible; a safe means for accomplishing this must be known. Secondly, if it becomes possible to produce fuel at the space station, thereby avoiding the launch costs involved with bringing it from Earth, a least time approach may become a least cost approach as well.

Pilots of high speed vehicles must possess proportionately quicker reflexes than those of slower craft. To dock a spacecraft in half the time requires a corresponding decrease in reaction time. Speed limits for automobiles were originally devised for safety reasons. The purpose of this investigation is to determine how initial (from 1000 ft) closing velocity affects a pilot's docking performance.

It should be mentioned that increased velocities are not obtainable without penalty. More fuel is required to accelerate to greater velocities and decelerate from them than from lower velocities. Also, orbital mechanics effects become more noticeable and significant at higher

velocities and may make dockings more difficult, possibly incurring even greater fuel consumption levels. These factors influence pilot performance. To avoid incurring large fuel costs, subjects are advised to maintain their maximum velocity for as long as feasible. In this way, it is possible to use virtually the same amount of fuel as in the NASA approach, but the Δv is applied more efficiently to result in less elapsed time.

Fuel consumption levels for the proposed manned orbital maneuvering vehicle were calculated assuming the use of shuttle fuel. The space shuttle has two means of maneuvering in orbit, the reaction control system (RCS) and the orbital maneuvering system (OMS). The OMS engines are used for Δv 's greater than 1.5 m/s, and have a specific impulse (Isp) of 313.2 s. The primary RCS (PRCS) engines are less efficient, with an Isp of 280.0 s. Calculations reveal that for a vehicle with a mass of 5000 kg, the PRCS requires 1.82 kg of fuel to produce a Δv of 1 m/s, while the OMS requires 1.63 kg of fuel for the same velocity increment. (See Appendix A.4.)

Propellant consumption and mission time are minimized when the + V-bar port is used for manned approaches. These factors make this location optimal for manned dockings. One important issue here is that of plume impingement. This is harmful both for surface contamination and transfer of momentum considerations. A "low-Z" mode is under consideration for the space shuttle. This successfully vectors the plumes away from the station, but consequently consumes twelve times more fuel than a nominal approach. Space station designers maintain that sensitive station components can be positioned or oriented relative to approach lanes to reduce these harmful impingement effects. Since the solar arrays rotate to follow the sun, an appropriately timed approach maneuver can be arranged to occur when the arrays are "edge-on" with respect to the impinging plume. This would be shortly before orbital noon. [NASA Johnson, 1984]

Vernon Larson and Stephen Evans describe a method for producing low cost fuel at the space station in a paper presented at the International Astronautical Federation in Innsbruck, Austria, 1986. Design goals for this space station propulsion method include high performance, extremely high reliability, long-life, controlled emissions and outgasses, and maximum safety. These goals are desirable and appropriate for a manned orbital vehicle as well. The proposed system produces fuel through water electrolysis of waste water from the laboratories and life support systems. The authors claim that once operational, this system will not require any transport of propulsion fluids from Earth; a specific impulse of 405 s, which is significantly better than both the OMS and RCS engines, has already been demonstrated. [Larson, 1986] If this system can be used on a vehicle of the type being simulated, a drastic change would be effected on the docking cost function, and a least time docking maneuver would clearly be a least cost solution as well.

4. Experimental Set-Up and Procedure

*For I dipt into the Future, far as human eye could see
Saw the Vision of the world, and all the wonder that would
be;
Saw the heavens fill with commerce, argosies of magic sails,
Pilots of the purple twilight, dropping down with costly
bales.
Alfred, Lord Tennyson*

Before detailing the experimental procedure, it is necessary to note the assumptions under which the experiment, and the software, were designed to operate.

- the fuel consumption and Δv are instantaneous
- the decrease in mass and change in center of mass due to fuel consumption are small enough to be omitted from calculations, allowing for a given impulse to produce the same Δv in all phases of the mission
- while thrusting, the Δv resulting from the thrusters is presumed to be much greater than the Δv arising from orbital mechanics effects
- the station is uniformly and continuously illuminated against a pitch black background
- it is more efficient to manually execute maneuvers than to compute the maneuvers using onboard targeting software. (Onboard computers may require up to two minutes to compute each and every necessary Δv using Lambert targeting and 10-15 seconds for Clohessy-Wiltshire targeting; each is only accurate for certain ranges. [NASA Johnson, 1983])

To determine a more appropriate approach profile, test subjects are initially acquainted with the simulator and are instructed as to the operation and function of the Digital Auto Pilot and the hand controllers. They are shown what will appear on the screen after completion of a successful docking mission. They are told to initially use HIGH acceleration for forward translation and proceed to LOW and PULSE as the range decreases. Orbital mechanics effects of thrusting are explained and test subjects are told to make the appropriate thrusts in the radial direction to maintain the image of the station on the screen. For approaches along the +V-bar, these radial thrusts are upward. The DAP configurations for radial motion are LOW and then PULSE. Subjects are instructed to strive for a quick approach without overthrusting, wasting fuel, or crashing. Essentially, the NASA protocol is followed with the removal of the velocity limits. Thruster commands are confined to two degrees of freedom.

A docking target is positioned on the space station such that when the spacecraft is on course the target is centered in the window and docking fixtures of chaser and target are properly aligned. By keeping this image in view, the pilot ensures a successful docking provided he/she correctly controls his/her final velocity.

The target is located at the space station's center of mass. It consists of a cyan circle with a radius of 3 m surrounding a "+" implemented with crosshairs. One meter in front of this (closer to

approaching craft) resides a 70% scale fixture in black. When the docking fixtures are aligned, the black target partially obscures the cyan target. (See Figure 4.1.) This configuration greatly simplifies the alignment problem.

Located at the docking port is a laser ranging system. This also facilitates docking by providing accurate range, range rate, angular position, and attitude of the chaser vehicle to the pilot. [NASA Johnson, 1984]

A successful docking is achieved by being in the right place at the right velocity. In the direction of orbital motion, the spacecraft must arrive between 1.5 and 2.0 m from the docking target at a rate of 0.05-0.15 m/s. In the other directions, the misalignment may be no greater than 0.23 m and the velocity must be less than 0.06 m/s. When this is done correctly, all vehicle motion automatically stops and mission costs are displayed. Upon incorrect docking, a "crash" routine runs which should be sufficiently unpleasant, both visually and aurally, to discourage further failures.

The chaser craft is initially positioned 304.8 m (1000 ft) from the target along the station's +V-bar at an altitude of 300 km. The station's image is visible on the main screen with flight data superimposed near the bottom. The testing is divided into three groups. In the first, the subject is instructed to complete two safe dockings at 0.3 m/s. This is the rate at which the NASA approach begins, and serves as a good introduction to the simulator.

The second group contains four sessions of ten runs each. Each session is characterized by a particular initial velocity ranging from 3.0 m/s to 9.0 m/s (a rate at which successful dockings were performed during preliminary investigations). The individual rates are 3.0, 5.0, 7.0, and 9.0 m/s.

The last group consists of ten runs at 3.0 m/s without the data displays. This group of runs is performed to determine importance of the range and range rate displays in a docking mission. Since these accurate data are provided by a laser docking system located at the docking target, they would not be available to an astronaut (or an automatic docking system) approaching a satellite, another spacecraft, or another part of the space station. Without these data, automatic docking is impossible, so this test is intended to reveal the utility and usefulness of manual control.

Values indicating pilot performance are mission time, cost, and fuel consumption. Y offset, y rate error, and z rate error are also recorded. Each test session lasts about 30 minutes.

Each experimental test subject is issued a training manual for perusal. (See Appendix A.5.) The test coordinator is available throughout the test sessions to answer questions and provide advice if necessary.

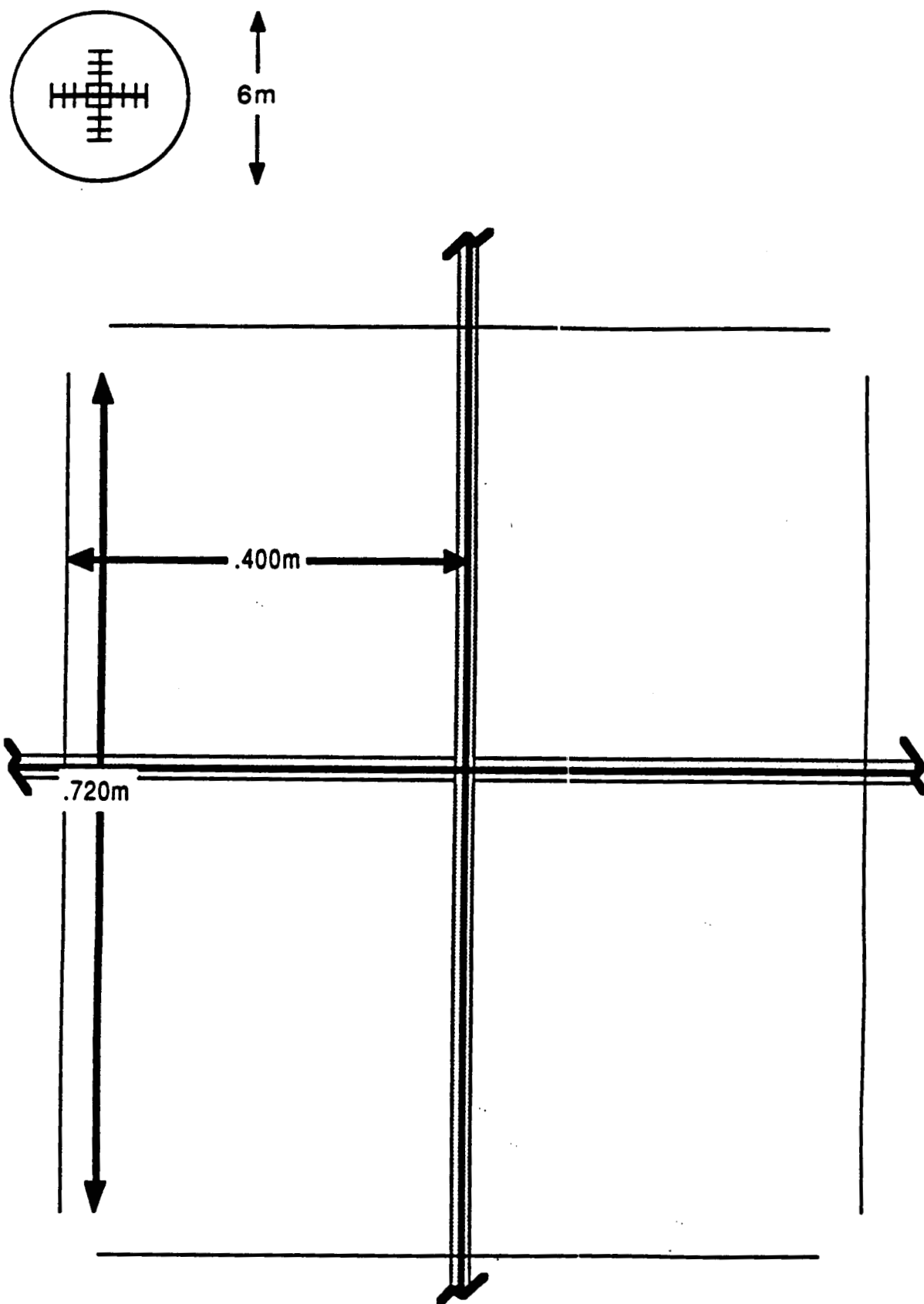


Figure 4.1: Docking Target and Docking Target (detail)

5. Simulator Evaluation

You ain't gonna learn what you don't want to know.
John Barlow, 1972

Although evaluation of the simulator as a training device is impossible without the benefit of feedback from someone who has performed an actual docking mission, comments and criticisms from test subjects pertaining to comfort, ease of use, and other hardware and software design issues should be taken under consideration and are worthy of mention here.

Several subjects became slightly uncomfortable in the zero-g chair after five missions. On orbit, they would more or less assume this position without the chair's assistance and would not suffer from sore knees, as on earth. Also, they would only perform one such mission at a time rather than ten in rapid succession, and so would not be as prone to other forms of fatigue either. Conducting these tests in a neutral buoyancy facility would solve the knee problem, but difficulties associated with wearing a mask and breathing apparatus and submerging the hardware might prove too cumbersome for the small gain in comfort.

Another hardware issue concerns the location of the DAP control panel. Some subjects operated the hand controller with the right hand at the beginnings of missions so as to minimize the delay between control mode selection and translation input. The DAP was located adjacent to the left hand controller for this very reason, but two-handed flight was used for further gains in performance. This minor inconvenience could be solved by either modifying the software to allow control mode selection before flight, or by relocating the DAP to the right side of the cockpit.

The IBM software limited the input sampling rate to about one per second. Although the graphics refresh rate was high enough for retinal image continuity, the hand controller had to be displaced for slightly less than 1 s for an input to be registered and sent. The test subjects quickly became accustomed to this delay to the point where it made little difference. In some cases, this delay prevented overthrusting, although in others, the sluggish response led to unsuccessful dockings.

Another control problem arose from the inability to send inputs for y translations smaller than 0.05 m/s. While this rate is acceptable and appropriate at medium to long ranges, fine tuning was sometimes difficult at small ranges, and pilot induced oscillations (PIO) were the result. One possible solution would be to reduce the y Δv inputs from 1.0, 0.10, and 0.05 m/s to, say, 0.50, 0.05, and 0.01. While this was considered during pre-experimental testing, it was thought that confusion would result from the disparity between the y and z rates. By installing an additional y pulse mode at a rate of 0.01 m/s rather than shifting all of the rates, this problem can be avoided. Another solution is to use the joystick as a proportional controller rather than merely a direction indicator.

In summation, all experimental shortcomings were minor to negligible, and are believed to have had little to no effect on the test results.

6. Experimental Results

*The more we learn, the less we believe to be true
The more we prove, the more remains to be proved.*
Peter Townshend, 1974

Ten students in MIT's Department of Aeronautics and Astronautics volunteered to be test subjects for this experiment. Selected test results appear as Figures 6.1-6.12. Raw data charts of time and fuel costs for each test subject are located in Appendix A.6. Figures 6.1 and 6.2 present average mission durations and average fuel consumptions for all subjects by initial velocity. Also indicated are the estimated values achieved by following the "0.1% rule". Total cost averages for all subjects appears as Figure 6.3.

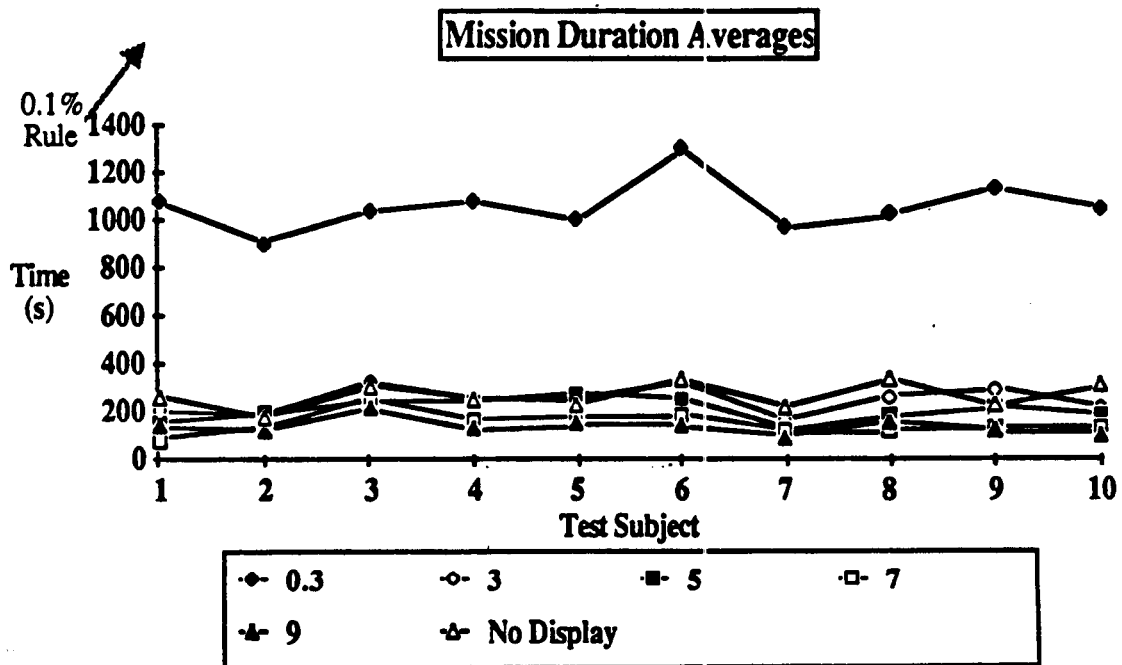


Figure 6.1: Mission Duration Averages

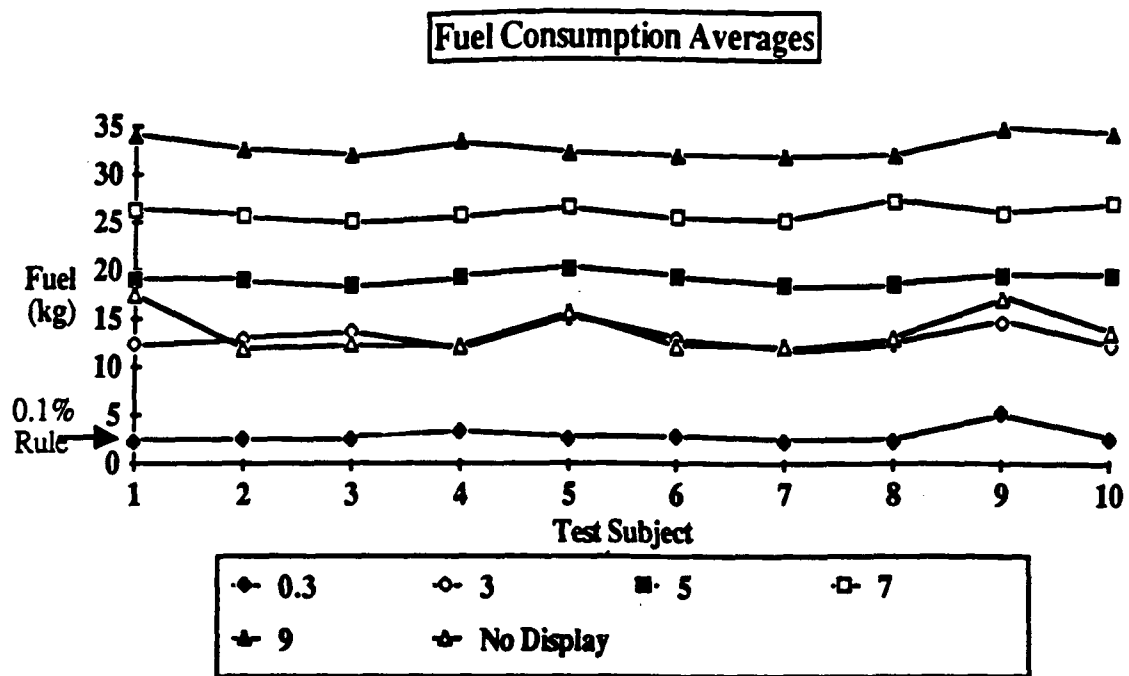


Figure 6.2: Fuel Consumption Averages

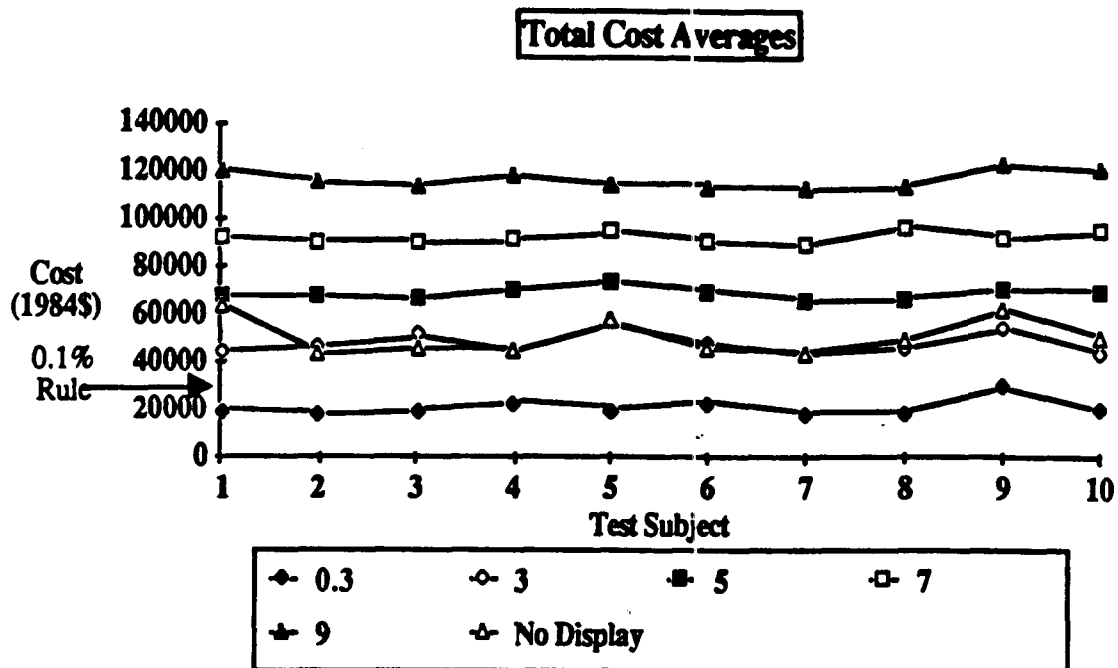


Figure 6.3: Total Cost Averages

By dividing the initial range by the initial velocity, a theoretical minimum mission duration is found. Any additional time ("convenience time") in the mission is attributable to the pilot's reaction time requirements. The minimum fuel consumption level, assuming no radial corrections, can also be calculated. These theoretical minimum values vs. initial velocity are plotted in Figure 6.4.

Theoretical Minimums Fuel (Starting fuel + Ending fuel) and Time

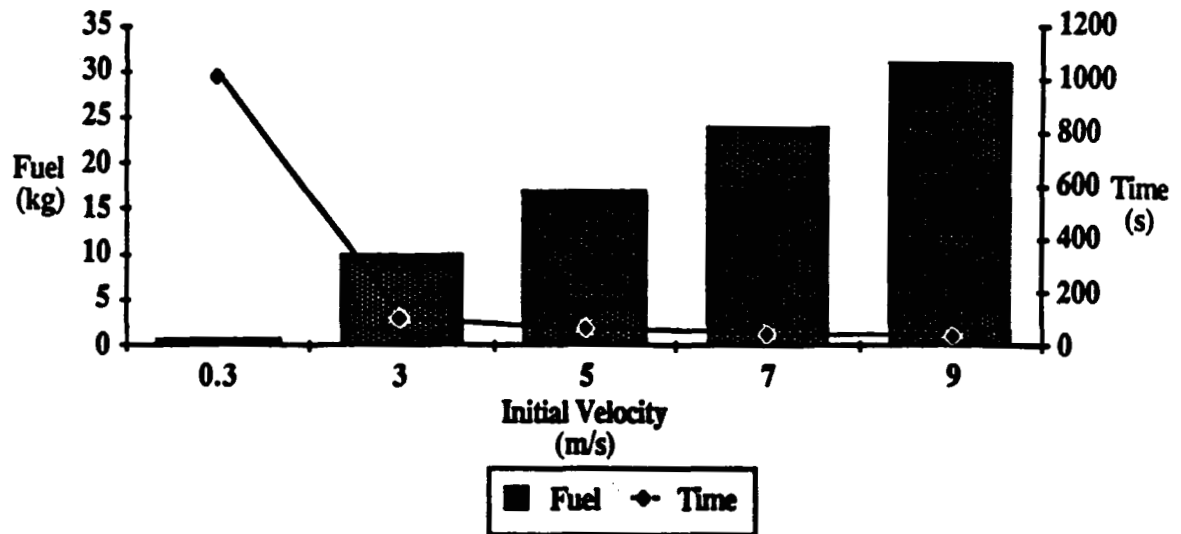


Figure 6.4: Theoretical Minimum Fuel and Time

Subtracting this minimum time from mission duration averages for all subjects yields Figure 6.5. Because of orbital mechanics effects, upward accelerations cause an increase in forward velocity. In several instances, this was used, intentionally or not, to a subject's advantage producing a negative "convenience" time.

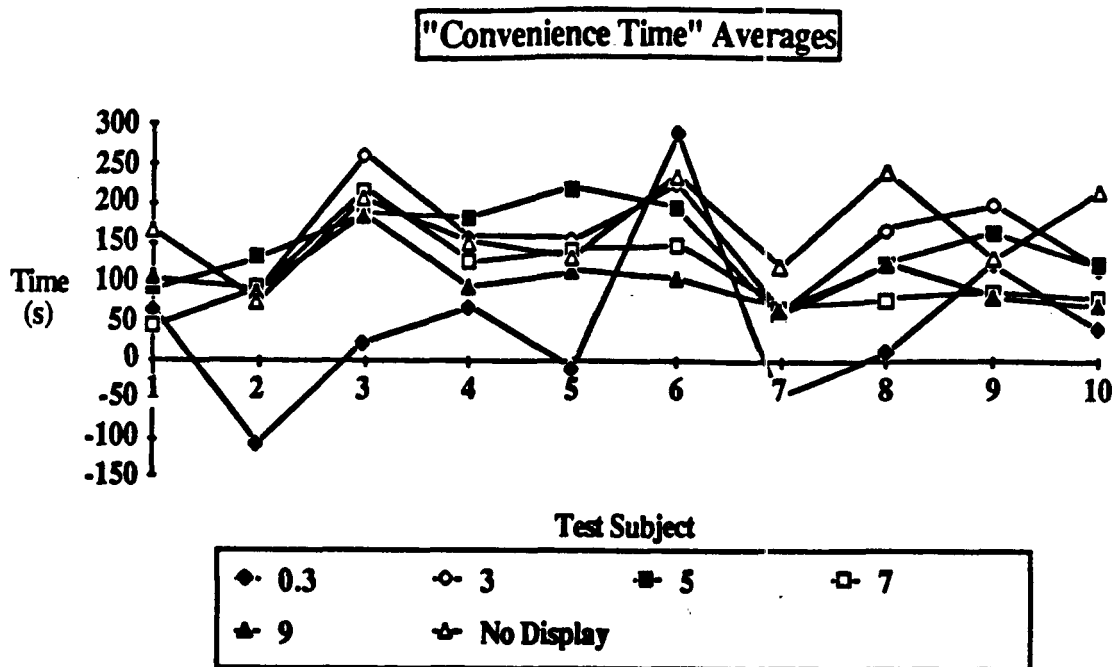


Figure 6.5: "Convenience Time" Averages

Radial fuel averages for all subjects appears as Figure 6.6. This indicates how initial velocity affects the amount of radial corrections.

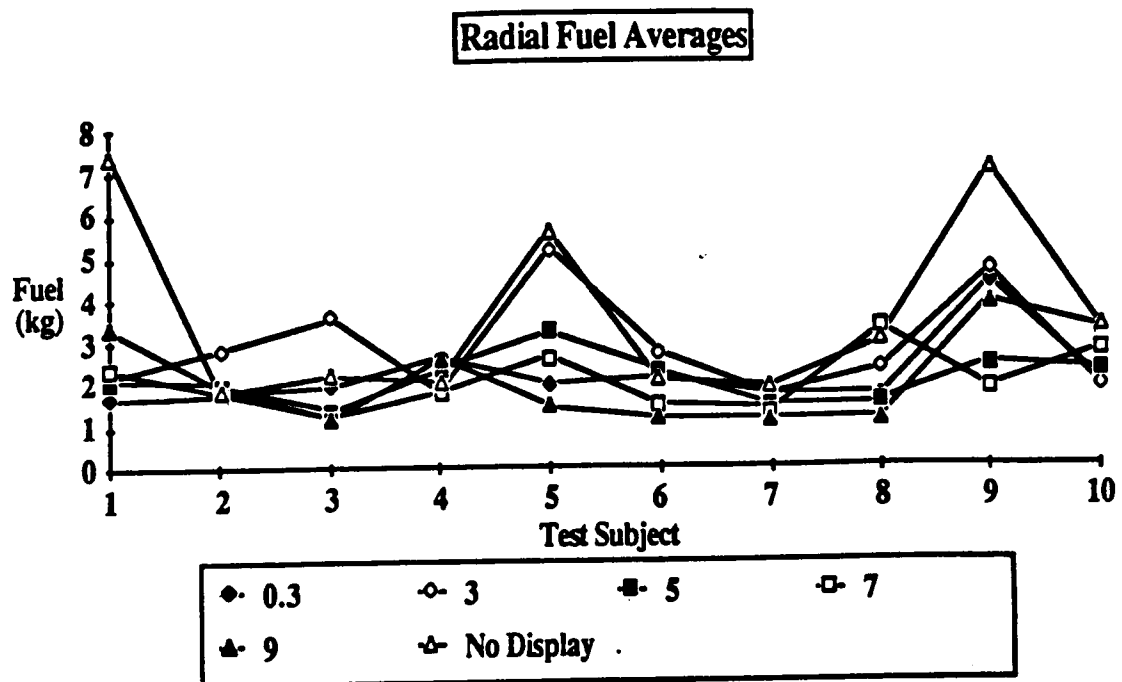


Figure 6.6: Radial Fuel Averages

A small proportion of the docking approaches were categorized as unsuccessful due to out-of-nominal final conditions. Many of the "crashes" would not have been catastrophic events in actuality but rather small errors in judgement leading to docking errors on the order of millimeters or millimeters per second. Technically, a minimum forward velocity of 0.05 meters per second must be met for a successful docking. All "crashes" registered because of insufficient velocity were discounted in the "No Display" data since it would not be a problem to add enough force to dock. This did not occur in any of the runs with the data displays. Figure 6.7 illustrates the number of unsuccessful docking attempts for all subjects by initial velocity.

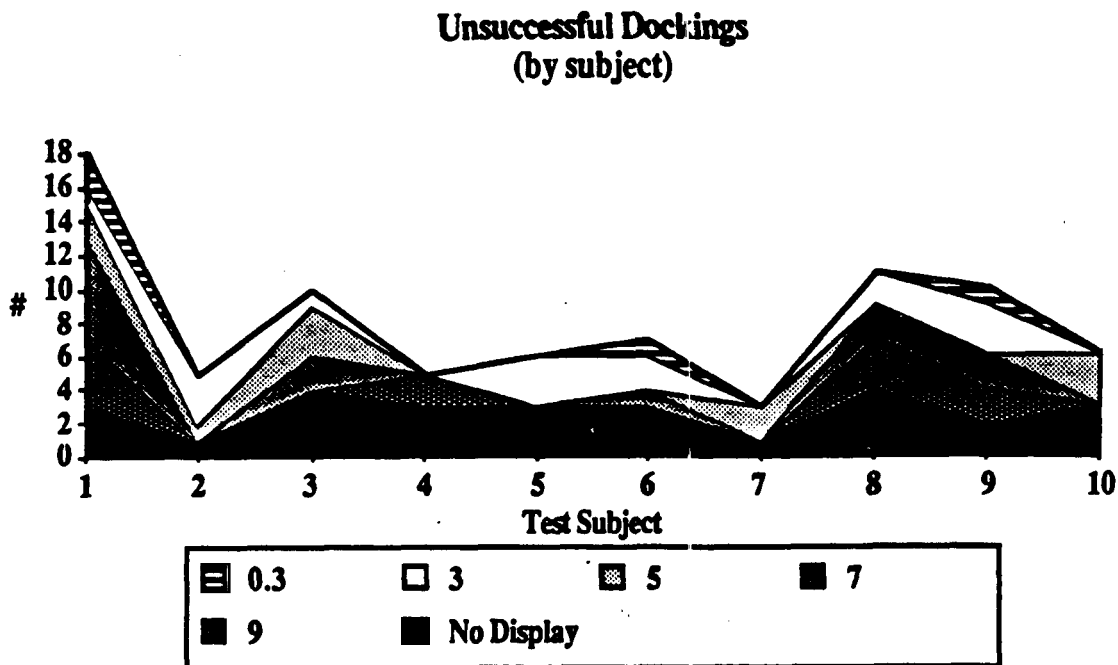


Figure 6.7: Unsuccessful Dockings (by subject)

Plotting mission duration and fuel consumption averages versus number of unsuccessful dockings produces Figures 6.8 and 6.9 respectively. The six subjects with fewer than 8 unsuccessful dockings can be analyzed separately to form graphs of mission duration averages, fuel consumption averages, and number unsuccessful. These data appear as Figures 6.10-6.12.

**Sum of Mission Duration Averages vs
Number of Unsuccessful Docking Attempts**

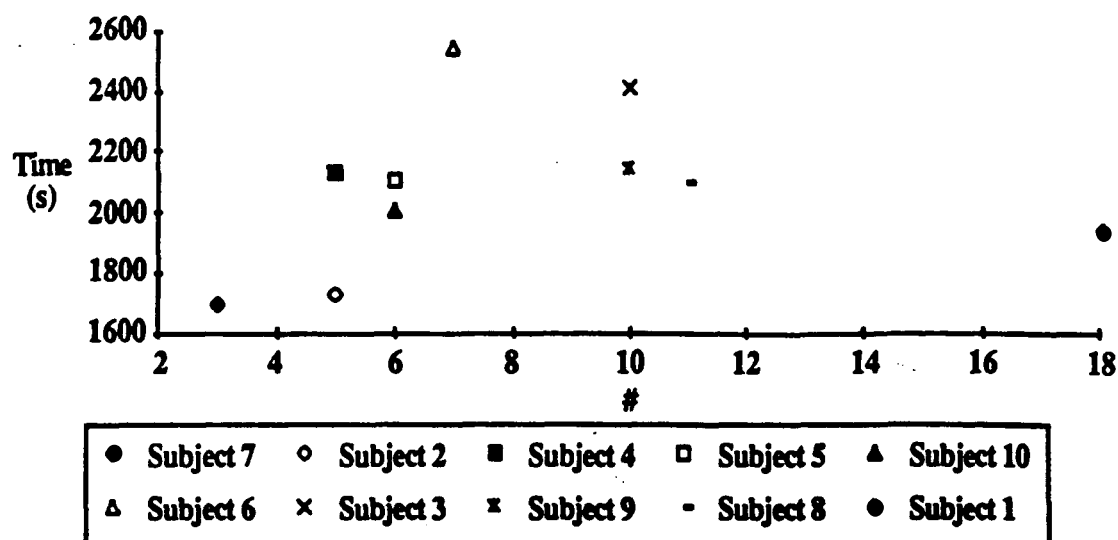


Figure 6.8: Sum of Mission Duration Averages vs. Unsuccessful Attempts

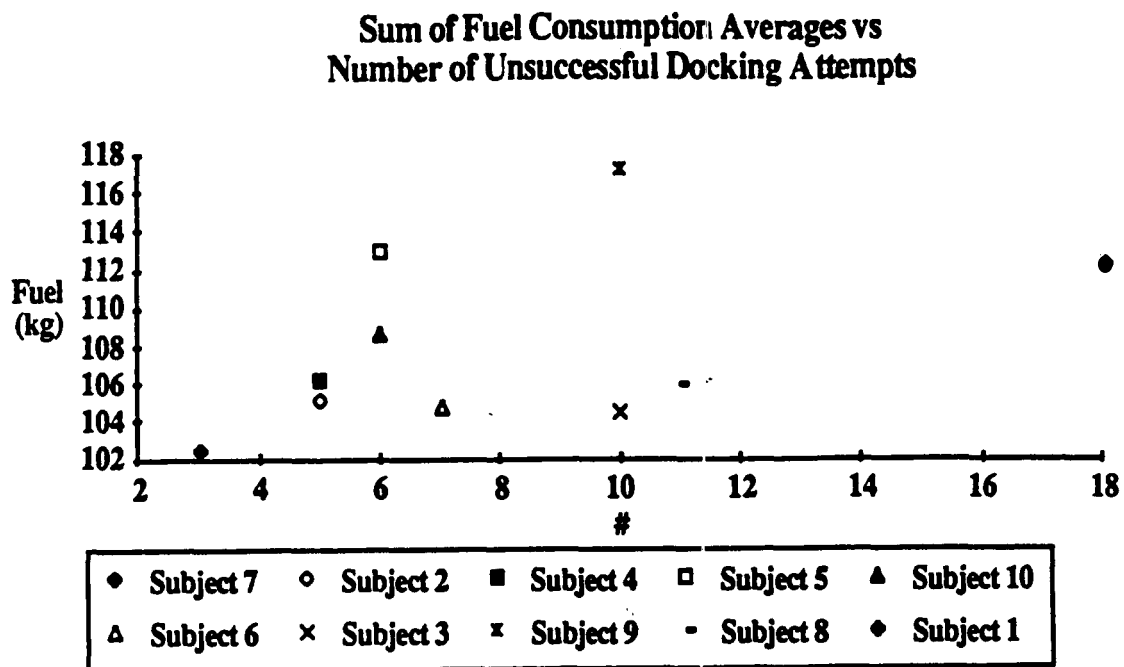


Figure 6.9: Sum of Fuel Consumption Averages vs. Unsuccessful Attempts

Mission Duration Averages (Best 6)

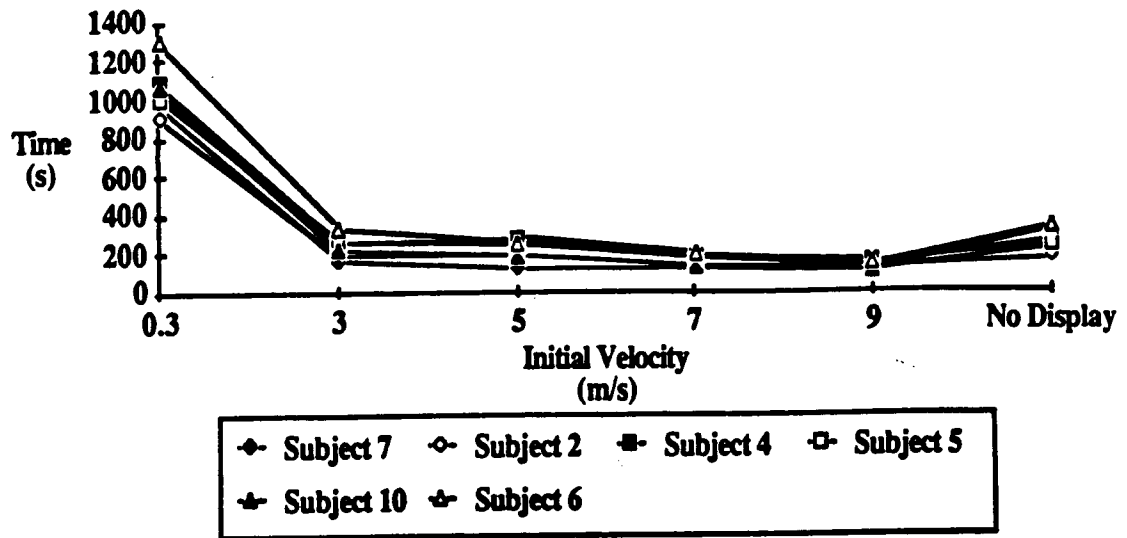


Figure 6.10: Mission Duration Averages for 6 Safest Subjects

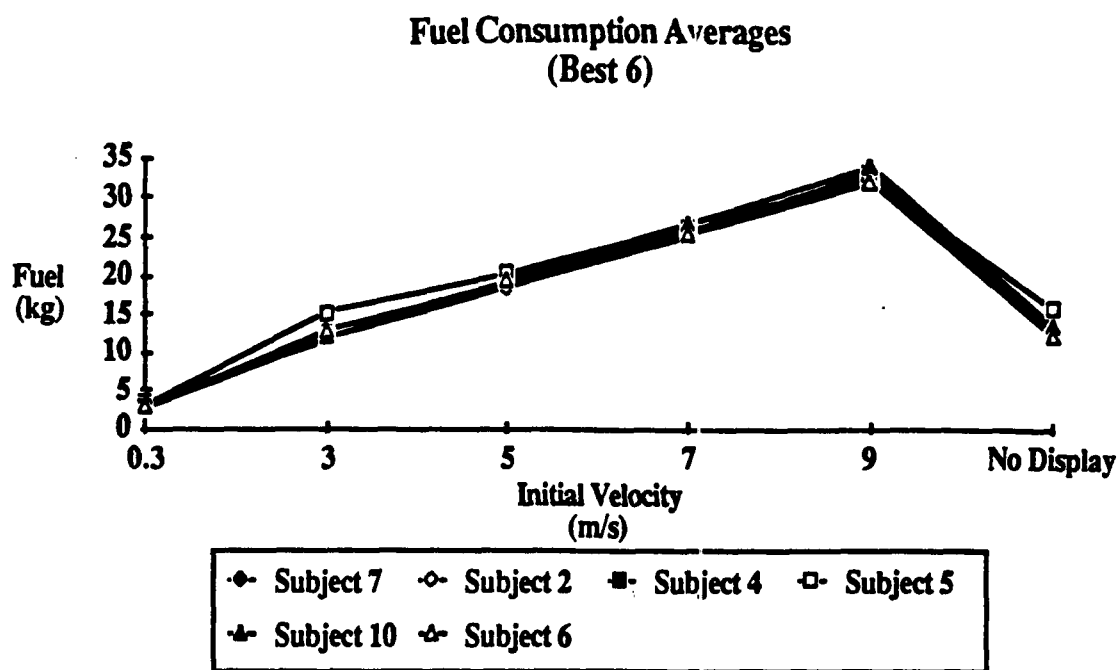


Figure 6.11: Fuel Consumption Averages for 6 Safest Subjects

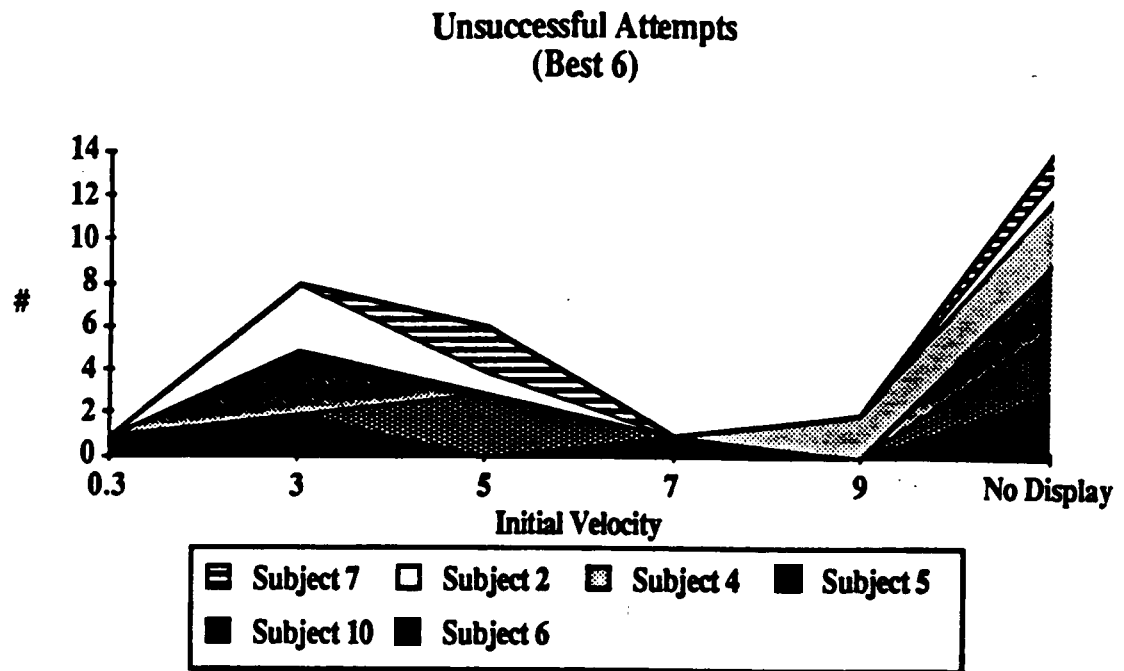


Figure 6.12: Unsuccessful Attempts for 6 Safest Subjects

7. Discussion--Analysis

*And all the science I don't understand,
It's just my job five days a week.
Bernie Taupin, 1972*

Before examining the results in any detail, it should be emphasized that all of the test subjects performed the required task without the benefit of computer-optimized trajectories. Computers can be used to calculate the optimal trajectory for any docking mission including least cost, least time, and least fuel, and even if they do not actually operate the thrusters, they can make suggestions as to their usage. None of the test subjects had access to the results of any such calculations, so the results are a good indication of unassisted manual control capabilities.

Also, while NASA stipulates that all space shuttle pilots have previous military pilot experience and subsequently trains them for several years before their first mission in space, none of the test subjects here had the benefit of either NASA or military training, pilot or otherwise. The results therefore indicate how individuals with minimal training might perform in a space docking operation.

Statistical analyses such as learning curves and data comparisons among subjects are not as valid here as they are with other experiments. In the design of an automobile, it is important to provide for safety for a wide and disparate range of operators. The safety of the worst "pilots" must be ensured. For space missions such as this one, the simulator defines the user population rather than vice versa. Subjects who perform well in the simulated mission will be given the opportunity to perform the actual mission. The flight system need not be modified (possibly reducing overall performance) to accommodate users on the low range of the performance scale. For this reason, examination of the best subjects' best performances is more appropriate than analysis of performance averaged over all subjects and over all runs. A comparison of averages over all runs and over all subjects is the most conservative way of analyzing the test data as the averages include both learning and fatigue but both sets of averages verify the conclusions.

The results indicate that test subjects with no more than a few minutes (as opposed to many years) practice can safely perform simulated spacecraft docking missions with approach velocities more than an order of magnitude greater than NASA would suggest for its military pilots performing comparable missions. While the cost of a crash would by far outweigh any marginal savings in time costs, sufficient training could reduce the probability of failure to almost any chosen design value.

Removing the data displays slightly raised the average mission duration (but did not make the mission impossible) in one-half of the subjects because they became more cautious without the feedback. This effect is shown more dramatically on "convenience time" averages.

If the protocol specified by NASA's "0.1 rule" is used as a baseline where the minimum mission time is 2683 s (45 minutes) and the estimated minimum cost is \$35,000, then it is possible to outperform this protocol both in time and cost while maintaining a high degree of safety. Costs were reduced by close to a factor of 2.

A comparison of mission duration averages over all subjects shows the unsurprising result that mission duration varies inversely with initial velocity. All times were at most one-half of what they would be were the "0.1%" rule followed and lowest times were 3.8% of the NASA profile

time. Eight of the subjects scored lowest average times with an initial velocity of 9 m/s. The times for 0.3 m/s are over ten minutes greater than the next highest times for every subject. One-half of the subjects scored second highest times at 3 m/s while the other half's second slowest scores were with the No Display runs. Removal of the data displays did not greatly affect docking time when averaged over all subjects.

Examination of the Fuel Consumption Averages chart (Figure 6.2) for all subjects reveals that fuel consumption is proportional to initial velocity for all subjects without exception. Removal of the data displays caused an increase in fuel consumption in 70% of the test subjects. This suggests that pilot uncertainty increases fuel usage. (See Figure 6.6.) Fuel data for the "0.1% rule" approach would be approximately equal to the data for an initial velocity of 0.3 m/s.

By beginning with the velocity mandated by the "0.1% rule", 0.3 m/s, all subjects accrued lower average costs than would be achieved by following the rule for the duration of the mission. By maintaining the initial velocity for a large part of the mission, the Δv was used more efficiently. While fuel costs at this rate are comparable to estimated values derived from the "0.1% rule", time costs were much less. The cost of fuel is so high that reduced time could not make up for increased fuel in any of the faster missions.

Plots of "convenience time" averages and radial fuel averages varied widely among subjects. Several clusterings of data points in each of the charts suggest that some subjects were able to keep these values constant, regardless of initial velocity. "Convenience time" averages were lowest with an initial velocity of 0.3 m/s for 70% of the subjects. Removal of the data displays caused the highest use of fuel for radial corrections for five of the test subjects. (See Figure 6.6.) An initial velocity of 9 m/s produced the lowest use of radial fuel for one-half of the subjects.

The number of unsuccessful dockings amounted to over 10% of the total runs for some subjects, but it must be remembered that these figures include learning and fatigue. Of the six safest subjects, only one subject had any failures (1) at 7 m/s and one subject registered 2 failures at 9 m/s. Removal of the data displays caused an 80% increase in the incidence of unsuccessful dockings over the 3 m/s runs with displays.

It is interesting to note from figures 6.8 and 6.9 that safety did not clearly increase with mission duration and fuel consumption as might be expected. In general, the subjects who docked with the lower durations and lower fuel consumption values were among the safest subjects.

Mission duration and fuel consumption charts for the six safest subjects indicate a linear dependence of time and fuel on velocity. Again, No Display values are similar to the 3 m/s values with displays.

8. Conclusions

I think my spaceship knows which way to go.
David Bowie

- If it is necessary to return to the space station because of an emergency, it is possible to do so much more quickly than the "0.1% rule" governs while still maintaining a high degree of safety.
- Docking costs can be substantially reduced by maintaining the initial velocity for most of the mission. These costs can be diminished tremendously if a fuel is produced on orbit as previously described.
- The "0.1% rule" for this type of vehicle is overly conservative.
- With sufficient training, accurate range and range rate data, such as those provided by a laser docking system are unnecessary, although helpful, for manual control. The resolution and accuracy of rendezvous radar should be sufficient for docking. In addition, radar is more versatile because it requires equipment only on the chaser spacecraft, although it is often used with a transponder on the target.
- There is no need for elaborate docking equipment on all targets.

9. Recommendations for Future Work

*poised for flight
wing spread
bright spring from
night into
the sun*
Robert Hunter, 1975

A virtually unlimited series of tests and experiments can be performed with a working flight simulator. Among these are investigations to determine preferred rendezvous trajectories and appropriate docking maneuvers for the space station or rotating satellites, and studies to find the best control modes, velocities, and accelerations for the above missions as well as any others. Comparisons with data derived from motion-based carriage, neutral buoyancy and other simulations would also be informative. Tests can be conducted to determine illumination needs, docking stresses, and requirements for ground station, time, ranging instrumentation and global positioning specifications. Abort modes and backup and failure modes can be explored. Audio cues such as thruster firings and alarms can be added as well as a voice synthesizer for additional inputs to the pilot.

The cockpit's hardware can be modified to unearth more possibilities. Controls can be added to operate docking fixtures, robotic arms, communications equipment, environmental regulators and clocks and timers. Programmable display pushbuttons can be added to assist with running launch and landing checklists. The current hardware can be reorganized to perform human factors tests of control station design, and control and display layouts. The installation of head-up displays and touch panel screens could also be advantageous.

The software can be modified to propose to the pilot alternative course-plotting algorithms for one-impulse, two-impulse, minimum time or minimum fuel trajectories. Additional graphics routines can be added to allow for growth of the station or approaches to other targets. Accurate star-field backgrounds and earth images can be added to achieve added realism. Experiments designed to provide for data loss contingencies can also be performed.

Ultimately, the simulator can be used as a training device for astronauts about to fly orbital missions. After the mission, the simulator's effectiveness as a training device, its impact on mission success, and its fidelity can be evaluated.

Appendix A.1--Computer Code

```

/* thesis2.c */
/* 11-10-86 */
#include "gl.h"
#include "device.h"
#include "math.h"
#include "stdio.h"
#include "flt_params.h"
#include "tipedefs.h"
#include "sys/types.h"
#include "sys/times.h"
#include "get.h"
#include "termio.h"
#include "fcntl.h"
#include "objects.c"
#include "picture.c"
#include "hills.c"
#include "impact.c"
#include "prt_data.c"
int crash,st;
char subj[30];
int star,up,back,data;
int input;
main()
(
    extern float conv();
    orbit3=pow(orbit,3.);
    n=sqrt(mu/orbit3);
    ginit();
    textport(0,400,0,50);
    color(BLUE);
    clear();
    printf("enter Test Pilot's name and rank: ");
    scanf("%s",subj);
    data_file=fopen(subj,"a");
    printf("enter initial z velocity: ");
    scanf("%f",&START_Z);

```

```

fprintf(data_file, "initial velocity=%.1f\n\n", START_Z);
fclose(data_file);
setmonitor(NTSC);
textport(40, 615, 10, 484);
viewport(0, 635, 0, 484);
ortho2(0., 633., 0., 484.);
for (st=1; st<30; st++)
    printf("\n");
printf("initial velocity=%.1f\n\n", START_Z);
cursoff();
trial=1;
mission_start();
}
mission_start()
{
    fuel_used=START_Z*1.63;
    doublebuffer();
    gconfig();
    tuttle();
    yaw_data=INITIAL_YAW;
    yaw_display=yaw_data;
    yaw_rad= -yaw_data*deg2rad;
    yaw_matr[0][0]=cos(yaw_rad);
    yaw_matr[0][2]= -sin(yaw_rad);
    yaw_matr[2][0]=sin(yaw_rad);
    yaw_matr[2][2]=cos(yaw_rad);
    x_range= INITIAL_X;
    y_range= INITIAL_Y;
    z_range= INITIAL_Z;
    zo=z_range;
    loadmatrix(identity);
    multmatrix(yaw_matr);
    getmatrix(cum_transform);
    editobj(SPACE_STATION);
        objreplace(NEWTAG);
        multmatrix(cum_transform);

```



```

        translate(-x_range,-y_range,-z_range);
    closeobj();
    tpoff();
    IBM=fopen("/dev/ttyd2","r");
    rate_z= START_Z;
    vzo=START_Z;
    vzon=vzo/n;
    x_rate=y_rate=0.;
    hill_time=0.;
    callobj(SPACE_STATION);
    prt_data();
    swapbuffers();

    st=0;
    while (st!=9)
        fscanf(IBM,"%d",&st);

    st=0;
    timer=times(&time_start);
    total_time=timcost=fuelcost=fuel_used=totalcost=0.;
    x_rate=y_rate=0.;
    while(trial<11)
    {
        if (fabs(z_range)<=2. && fabs(y_range)<2. && fabs(x_range)<2.)
        {
            if (fabs(z_range)<1.5 || fabs(y_range)>.23 || fabs(x_range)>.23 ||
                fabs(rate_z)>.15 || fabs(y_rate)>.06 || fabs(x_rate)>.06)
            {
                impact();
                crash=TRUE;
                printf("**crash**\n");
                mission_end();
            }
            else if (fabs(rate_z)>=.05)
                mission_end();
        }

        fscanf(IBM,"%d",&input);
        star=floor(input/100.);
    }

```

```

up=floor((input-(star*100.))/10.);
back=input-(star*100.)-(up*10.);
starboard=conv(star);
upward=conv(up);
backward=conv(back);
    fuel_used+=fabs(starboard)*TRANS_FUEL+fabs(upward)*
        TRANS_FUEL+fabs(backward)*TRANS_FUEL;

time2=timer;
time=times(&time_start);
time_inc_in60ths=timer-time2;
picture();
}
mission_end()
(
    fflush(IBM);
    tpon();
    color(BLUE);
    clear();
    timecost=total_time*TIME_COST;
    fuelcost=fuel_used*FUEL_COST;
    totalcost=timecost+fuelcost;
    printf("trial %d\n",trial);
    data_file=fopen(subj,"a");
    if (crash==TRUE)
    (
        printf("**crash**\n");
        fprintf(data_file,**crash**\n");
    )
    crash=FALSE;
    printf("final y= %.2f  final y_rate= %.2f  final z_rate= %.2f\n",
        y_range,y_rate,rate_z);
    printf("total_time= %.2f s\t time cost= $ %.2f\n",total_time,timecost);
    printf("total_fuel= %.2f kg\t fuel cost= $ %.2f\n",fuel_used,fuelcost);
    printf("\t\ttotal cost= $.2f\n",totalcost);
    swapbuffers();

```

```

fprintf(data_file,"trial %d\n",trial);
fprintf(data_file,"final y= %.2f  final y_rate= %.2f",
        y_range,y_rate);
fprintf(data_file,"  final z_rate= %.2f\n",rate_z);
fprintf(data_file,"total_time= %.2f s\t time cost= $ %.2f\n",
        total_time,timecost);
fprintf(data_file,"total_fuel= %.2f kg\t fuel cost= $ %.2f\n",
        fuel_used,fuelcost);
fprintf(data_file,"t\ttotal cost= $%.2f\n\n",totalcost);
fclose(data_file);
while (st!=9)
    fscanf(IBM,"%d",&st);
st=0;
trial++;
mission_start();
}
float conv(data)
{
    if (data==8.) /* 1.0 */
        return(.1);
    else if (data==6.) /* 0.1 */
        return(.01);
    else if (data==2.) /* 0.05 */
        return(.005);
    else if (data==1.) /* -0.05 */
        return(-.005);
    else if (data==3.) /* -0.1 */
        return(-.01);
    else if (data==4.) /* -0.05 */
        return(-.1);
    else
        return(0.);
}

```

```

/* flt_params.h */
/* This file contains all flight control parameters for
   Adam Brody's thesis work. It is to be included with
   all flight control systems. */

/* 10-15-86 */
#define HIGH_TRANS .1          /* m/s */
#define LOW_TRANS .05
#define TRANSLATE_PULSE .01

#define ROTATE_PULSE .05      /* deg/s */

#define TRANS_FUEL 1.82        /* PRCs fuel consumption in kg/m/s */
#define ROTATE_FUEL .01        /* fictitious value fuel consumption in kg/deg/s */
#define TIME_COST 10.16        /* 1984$/s for 2 astronauts */
#define FUEL_COST 3500         /* 1984$/kg */

#define mu 398604.              /* km^3/s^2 */
#define orbit 6674.            /* station orbit in km */
#define EARTH_RAD 6378000.     /* m */

#define INITIAL_X 0.
#define INITIAL_Y 0.
#define INITIAL_Z -304.8
#define INITIAL_YAW 180.

#define X.CG_OFFSET
#define Y.CG_OFFSET
#define Z.CG_OFFSET

```

```

/* tipedefs.h
here are all type definitions
9-12-86 */

float START_Z;
int trial;
FILE *IBM,*data_file;
float initial_x_rate,initial_y_rate;
float roll,pitch,yaw;
float roll_rad,pitch_rad,yaw_rad; /* degrees in radians */
float cos_roll,sin_roll,cos_pitch,sin_pitch,cos_yaw,sin_yaw;
float cos_roll_data,cos_pitch_data,cos_yaw_data;
float sin_roll_data,sin_pitch_data,sin_yaw_data;
float time_inc_in60ths,real_time_inc,time2,time_start,timer;
float trans_time,rot_factor;
float x_range,y_range,z_range; /* data displays */
float x_rate,y_rate;
float rate_z,rate_roll;
float pitch_rate,yaw_rate;
float roll_display,pitch_display,yaw_display; /* data displays */
float x_range_data,y_range_data,z_range_data;
float x_rate_data,y_rate_data,z_rate_data;
float roll_data,pitch_data,yaw_data; /* data for translations */
float starboard,upward,backward; /* rates in vehicle coords */
short stay_in_hills;
short start_hills;
float n_orbit3; /* time since last translation */
float hill_time;
float nhill_time;
float total_time;
float non_hills_delta_x,non_hills_delta_y,non_hills_delta_z;
float xo,yo,zo,vzo;
float vxon,vyon,vzon,csnt,snnt; /* save cpu time */
float fuel_used;
Matrix transform,dummy;
char ranges[40]; /* create character matrix for data*/

```

```

char range_rates[40];
char angles[40];
char angle_rates[40];
char fuel[20];
char fuel_cost[20];
char time_char[20];
char time_cost[20];
char total_cost[40];
float fuel_cost, timecost, totalcost;
float identity[4][4]=

      ((1.,0.,0.,0.),
       ( 0.,1.,0.,0.),
       ( 0.,0.,1.,0.),
       ( 0.,0.,0.,1.));

float roll_matr[4][4]=

      ((1.,0.,0.,0.),
       ( 0.,1.,0.,0.),
       ( 0.,0.,1.,0.),
       ( 0.,0.,0.,1.));

float pitch_matr[4][4]=

      ((1.,0.,0.,0.),
       ( 0.,1.,0.,0.),
       ( 0.,0.,1.,0.),
       ( 0.,0.,0.,1.));

float yaw_matr[4][4]=

      ((1.,0.,0.,0.),
       ( 0.,1.,0.,0.),
       ( 0.,0.,1.,0.),
       ( 0.,0.,0.,1.));

float cum_transform[4][4]=

```

```

/* objects.c */
/* for use with editobj(): */
/* 8-19-86 */
/* pruning 9-22-86 */
#include "gl.h"
#define DOCKING_PORT 1
#define DUAL_KEEL 2
#define PANEL_SUPPORTS 3
#define OUTER_TRUSS 4
#define TRUSS 5
#define FLAG 6
#define MIT 7
#define PANELS 8
#define CROSSPIECE 9
#define SPACE_STATION 10
#define EARTH 11
extern Matrix transform;
Tag NEWTAG;
int z,m,nl,a;
short as;
Angle angle;
float b,x;
#define DOCKING_PORT_RADIUS 3
#define DOCKING_PORT_GUIDE 1.6
tuttle()
(
int bigside=16;
makeobj(DUAL_KEEL);
bbox2(1,1,-10.,-100.,10,60.);
color(BLUE);
recti(-bigside/2,52,bigside/2,-98);
recti(-6,50,6,-96);
closeobj();

makeobj(DOCKING_PORT);
bbox2(1,1,-4.,-4.,4.,4.);

```

```

rectf(-DOCKING_PORT_GUIDE,-.01,DOCKING_PORT_GUIDE,.01);
rectf(-.01,-DOCKING_PORT_GUIDE,.01,DOCKING_PORT_GUIDE);
circi(0,0,DOCKING_PORT_RADIUS);
pushmatrix();
for (as=1;as<3;as++)
{
    for (b=-DOCKING_PORT_GUIDE;b<=DOCKING_PORT_GUIDE;b+=.4)
    {
        move2(b,.36);
        rdr2(0,-.72);
    }
    rotate(900,'z');
}
popmatrix();
closeobj();

makeobj(PANEL_SUPPORTS);
bbox2(1,1,-50.,-25.,50.,40.);
color(BLUE);
for(z=0;z<2;z++)
{
    move2i(-48+20*z,37);
    draw2i(-48+20*z,-21);

    move2i(28+20*z,37);
    draw2i(28+20*z,-21);
}
closeobj();

makeobj(OUTER_TRUSS);
color(BLUE);
pushmatrix();
movei(bigside/2,52,bigside/2); /*four corners*/
drawi(bigside/2,52,-bigside/2);
movei(bigside/2,-98,-bigside/2);
drawi(8,-98,8);

```



```

movei(-8,-98,8);
drawi(-8,-98,-8);
movei(-8,52,8);
drawi(-8,52,-8);
popmatrix();
for(b=0.;b<=16.;b=b+16.)
{
    pushmatrix();
    translate(b-8.,0.,0.);
    callobj(TRUSS);
    popmatrix();
}
closeobj();

makeobj(TRUSS);
bbox2(1,1,-5.,30.,5.,55.);
for(z=0;z<=135;z=z+15)
{
    color(MAGENTA);
    movei(0,z-98,bigside/2);
    drawi(0,z-83,-8);
    drawi(0,z-83,bigside/2);
}
closeobj();

makeobj(FLAG);
bbox2(1,1,-10.,-5.,8.,10.);
color(BLUE); /*blue field*/
rectf(0.,8.,4.,1.);
color(WHITE);
for(nl=0;nl<=4;nl++) /*rows with six stars*/
{
    for (m=1;m<=6;m=m+1)
    {
        pnt2(4.*m/7.,189./26.-9.*nl/13.);
    }
}

```

```

)
for(n1=0;n1<=3;n1++)
{
    for(m=1;m<=5;m++)
        pnt2(2./7.+4*m/7.,180./26.-9.*n1/13.);
}
color(RED);
for(n1=0;n1<=3;n1++)
    rectf(4.,8.-n1*16./13.,12.,96./13.-n1*16./13.);
for(n1=0;n1<=2;n1++)
    rectf(0.,40./13.-n1*16./13.,12.,32./13.-n1*16./13.);

color(WHITE);
for(n1=0;n1<=2;n1++) {
    rectf(4.,96./13.-n1*16./13.,12.,88./13.-n1*16./13.);
}
for(n1=0;n1<=2;n1++) {
    rectf(0.,48./13.-n1*16./13.,12.,40./13.-n1*16./13.);
}
closeobj();

makeobj(MIT); /* 1-17*/
bbox2(1,1,0.,-5.,6.,18.);
color(GREEN); /* 'T' */
move2i(3,0);
draw2i(3,5);
move2i(1,5);
draw2i(5,5); /* 'I' */
move2i(2,6);
draw2i(4,6);
move2i(3,6);
draw2i(3,11);
move2i(2,11);
draw2i(4,11); /* 'M' */
move2i(1,12);

```

```

draw2i(1,17);
draw2i(3,15);
draw2i(5,17);
draw2i(5,12);

closeobj();
makeobj(SPACE_STATION);
color(BLACK);
clear();
perspective(200,1.33,0.,1000.);
maketag(NEWTAG);
multmatrix(identity);
    translate(dummy);
pushmatrix();
translate(0.,-orbit*1000.,0.);
callobj(EARTH);
popmatrix();
    callobj(PANEL_SUPPORTS);
    callobj(OUTER_TRUSS);
    callobj(CROSSPIECE);
    callobj(PANELS);
    translate(0.,0.,-8.);
        callobj(DUAL_KEEL);

pushmatrix();
    translate(0.,0.,7.);
    scale(.7,.7,.7);
        color(BLACK);
        callobj(DOCKING_PORT);

popmatrix();
    translate(0.,0.,8.); /* midpoint */
    color(CYAN);
    callobj(DOCKING_PORT);

pushmatrix();
    translate(0.,0.,1.);
    scale(.7,.7,.7);
        color(BLACK);
        callobj(DOCKING_PORT);

```

```

popmatrix();
  translate(0.,0.,8.);
    callobj(DUAL_KEEL);
  translate(-6.,-70.,0.);
    callobj(FLAG);
  translate(3.,15.,0.);
    callobj(MIT);

closeobj();
makeobj(CROSSPIECE); /*lateral support*/
bbox2(1,1,-50.,5.,50.,10.);
  color(RED);
  recti(-48,9,48,7); /* 8m above cm*/
  pushmatrix();
  for(a=0;a<96;a=a+3) (
    move2i(a-48,7);
    draw2i(a-45,9);
    draw2i(a-45,7);
  )
  popmatrix();
closeobj();

makeobj(PANELS);
bbox2(1,1,-50.,-15.,50.,30.);
  solar_panel(-48,26);
  solar_panel(-48,-10);
  solar_panel(-28,26);
  solar_panel(-28,-10);

  solar_panel(28,26);
  solar_panel(28,-10);
  solar_panel(48,26);
  solar_panel(48,-10);
closeobj();

makeobj(EARTH);
rotate(900,'x');

```

```

bbox2(1,1,-35000000.,-35000000.,35000000.,35000000.);
mapcolor(42,0,84,120);
mapcolor(43,0,5,5);
color(42);
    for (angle=1;angle<=4;angle++)
    {
        circf(0.,0.,EARTH_RAD);
        rotate(900,'x');
    }

color(43);
for (z=1;z<=3600;z+=200)
{
    circ(0,0,EARTH_RAD);
    rotate(200,'y');
}
rotate(900,'z');
for (z=0;z<=3600;z+=200)
{
    circ(0,0,EARTH_RAD);
    rotate(200,'y');
}
closeobj();
solar_panel(x,y,s)
int x,y;
int s;
{
    color(YELLOW);
    recti(x-3,y+11,x+3,y-11);
    for(s=y-10;s<=y+10;s++) /*cross spokes*/
    {
        move2i(x-3,s);
        draw2i(x+3,s);
    }
}

```

```

/* picture.c
   this program designates thruster commands from the
   keyboard for rotate and translate and accel control
   11-10-86 */
#define rad2deg 57.295779 /* 180./PI */
#define deg2rad .017453292 /* PI/180. */
picture()
{
    real_time_inc=time_inc_in60ths/60.;
    rot_factor=deg2rad*real_time_inc;

    roll_rad=roll*rot_factor;
    yaw_rad=yaw*rot_factor;
    pitch_rad=pitch*rot_factor;

    cos_roll=cos(roll_rad);
    sin_roll=sin(roll_rad);
    cos_pitch=cos(pitch_rad);
    sin_pitch=sin(pitch_rad);
    cos_yaw=cos(yaw_rad);
    sin_yaw=sin(yaw_rad);

    roll_matr[0][0]=cos_roll;
    roll_matr[0][1]=sin_roll;
    roll_matr[1][0]=-sin_roll;
    roll_matr[1][1]=cos_roll;

    pitch_matr[1][1]=cos_pitch;
    pitch_matr[1][2]=sin_pitch;
    pitch_matr[2][1]=-sin_pitch;
    pitch_matr[2][2]=cos_pitch;

    yaw_matr[0][0]=cos_yaw;
    yaw_matr[0][2]=-sin_yaw;
    yaw_matr[2][0]=sin_yaw;
    yaw_matr[2][2]=cos_yaw;

```

```

cos_roll_data=cos(roll_data*deg2rad);
sin_roll_data=sin(roll_data*deg2rad);
cos_pitch_data=cos(pitch_data*deg2rad);
sin_pitch_data=sin(pitch_data*deg2rad);
cos_yaw_data=cos(yaw_data*deg2rad);
sin_yaw_data=sin(yaw_data*deg2rad);
if (starboard!=0. || upward!=0. || backward!=0.)
{
    hill_time=0.;
    /* inertial/vehicle coords */
    xo=x_range;
    yo=y_range;
    zo=z_range;

    initial_x_rate=cos_yaw_data*cos_roll_data*starboard+
        (sin_pitch_data*sin_yaw_data*cos_roll_data-
         cos_pitch_data*sin_roll_data)*upward+
        (cos_pitch_data*sin_yaw_data*cos_roll_data+
         sin_pitch_data*sin_roll_data)*backward+x_rate;

    initial_y_rate=cos_yaw_data*sin_roll_data*starboard+
        (cos_pitch_data*cos_roll_data+sin_pitch_data*
         sin_yaw_data*sin_roll_data)*upward-
        (cos_pitch_data*sin_yaw_data*sin_roll_data-
         sin_pitch_data*cos_roll_data)*backward+y_rate;

    vzo= -sin_yaw_data*starboard+
        sin_pitch_data*cos_yaw_data*upward+
        cos_pitch_data*cos_yaw_data*backward+rate_z;
    vxon=initial_x_rate/n;
    vyon=initial_y_rate/n;
    vzon=vzo/n;
}
hills();
/* change back to vehicle coords */
loadmatrix(identity);

```

```

multmatrix(roll_matr);
multmatrix(pitch_matr);
multmatrix(yaw_matr);
multmatrix(cum_transform);
getmatrix(cum_transform);

editobj(SPACE_STATION);
  objreplace(NEWTAG);
  multmatrix(cum_transform);
  translate(-x_range,-y_range,-z_range);
closeobj();

callobj(SPACE_STATION);
prt_data();
swapbuffers();
total_time+=real_time_inc;
roll_pitch=yaw=0.; /* rate control */
starboard=upward=backward=0.;
rate_roll=pitch_rate=yaw_rate=0.;
}

```



```

/* hills.c */
/* this program performs orbital mechanics computations */
/* 9-17-86 */
/* Hill's IRIS
   x      y      radial
   y      -z     forward
   z      -x     port side
*/
hills()
{
    nhill_time=n*hill_time;
    csnt=cos(nhill_time);
    snnt=sin(nhill_time);

    if (hill_time==0.)
    {
        non_hills_delta_x=initial_x_rate*real_time_inc;
        non_hills_delta_y=initial_y_rate*real_time_inc;
        non_hills_delta_z=vzo*real_time_inc;
    }

    else
    {
        non_hills_delta_x=0.;
        non_hills_delta_y=0.;
        non_hills_delta_z=0.;
    }

    /* in space station! (not hills or vehicle) coords */
    y_range=non_hills_delta_y+vyon*snnt+(2*vzon-3*yo)*csnt+(2*(-vzon)+4*yo);
    z_range=non_hills_delta_z+ -2*vyon*csnt+(4*vzon-6*yo)*
        snnt+z0+2*vyon+(3*(-vzo)+6*n*yo)*hill_time;
    x_range=non_hills_delta_x+xo*csnt+vxon*snnt;

    y_rate=initial_y_rate*csnt-(2*vzo-3*n*yo)*snnt;
    rate_z=2*initial_y_rate*snnt+(4*vzo-6*n*yo)*csnt-3*vzo+6*n*yo;
    x_rate= -xo*n*snnt+initial_x_rate*csnt;
    hill_time=real_time_inc+hill_time; /* hill_time=time since translation */
}

```

```

/* impact.c */
/* This is an obnoxious crash routine to discourage subjects from crashing */

impact()
{
    int stall;
    float sint;
    frontbuffer(TRUE);
    ortho2(-.5,1067.5,-.5,767.5);
    for (sint=1.;sint<=100.;sint++)
    {
        color(BLUE);
        circ(200.,600.,sint/10.);
        circ(700.,300.,sint*2.);

        color(YELLOW);
        circ(530.,380.,sint/2.);
        circ(300.,250.,sint-5.);

        color(RED);
        circ(1000.,700.,sint/3.-5.);
        circ(480.,20.,sint);

        color(MAGENTA);
        circ(250.,650.,(sint*2.)-40.);
        rect(30.+sint/10.,300.+sint/10.,50.-sint/10.,400.-sint/10.);

        color(GREEN);
        circ(760.,700.,sint);
        circ(500.,250.,sint-50.);

        color(CYAN);
        circ(10.,10.,sint*3.);
        rect(500.+sint/2.,300.+sint/2.,600.-sint/2.,400.-sint/2.);

        color(WHITE);
        circ(1000.,20.,sint-100.);
        circ(25.,395.,-sint);

        for (stall=1;stall<200;stall++)
        {
            ringbell();
        }
    }
}

```

```

sleep(1);
setbell(2);
ringbell();
move2i(XMAXSCREEN, YMAXSCREEN);
color(CYAN);
for (stall=1;stall<=3300;stall++)
{
    draw2i(650,500);
    rotate(stall,'z');
    if (stall<120)
        ringbell();
    else
        for (sint=1.;sint<110.;sint++)
            {}
    exit(0);
}

```

```

/* prt_data.c */
/* 9-19-86 */
prt_data()
{
    roll_data= rad2deg*atan2(cum_transform[1][0],cum_transform[0][0]);
    pitch_data= rad2deg*atan2(cum_transform[2][1],cum_transform[2][2]);
    yaw_data= rad2deg*asin(-cum_transform[2][0]);

    if (yaw_data>=90.)
        yaw_data-=90.;
    else if (yaw_data<= -90.)
        yaw_data+=90.;

    roll_display-=roll*real_time_inc;
    pitch_display-=pitch*real_time_inc;
    yaw_display-=yaw*real_time_inc;

    if (roll_display>360.)
        roll_display-=360.;
    else if (roll_display<-360.)
        roll_display+=360.;
    if (pitch_display>360.)
        pitch_display-=360.;
    else if (pitch_display<-360.)
        pitch_display+=360.;
    if (yaw_display>360.)
        yaw_display-=360.;
    else if (yaw_display<-360.)
        yaw_display+=360.;

    pushmatrix();
    ortho2(-.5,1067.5,-.5,767.5);

    sprintf(ranges,"x range = %4.2f ",-x_range); /* create str */
    sprintf(range_rates,"x rate = %4.2f", -x_rate);
    cmov2s(95,90);

```

```

if (x_range<-.23)
    color(GREEN);
else if (x_range>.23)
    color(YELLOW);
else
    color(RED);
charstr(ranges); /* print str */

    cmov2s(380,90);
if (x_rate<-.06)
    color(GREEN);
else if (x_rate>.06)
    color(YELLOW);
else
    color(RED);
charstr(range_rates);

    sprintf(ranges,"y range = %4.2f",y_range);
    sprintf(range_rates,"y rate = %4.2f",y_rate);
    cmov2s(95,70);
if (y_range<-.23)
    color(GREEN);
else if (y_rate>.23)
    color(YELLOW);
else
    color(RED);
charstr(ranges);

    cmov2s(380,70);
if (y_rate<-.06)
    color(GREEN);
else if (y_rate>.06)
    color(YELLOW);
else
    color(RED);

```

```

charstr(range_rates);

    printf(ranges,"z range = %4.2f",z_range);
    printf(range_rates,"z rate = %4.2f",rate_z);
    cmov2s(95,50);
    if (z_range<-2.)
        color(GREEN);
    else if (z_range>-1.5)
        color(YELLOW);
    else
        color(RED);
    charstr(ranges);

    cmov2s(380,50);
    if (rate_z<.05)
        color(GREEN);
    else if (rate_z>.15)
        color(YELLOW);
    else
        color(RED);
    charstr(range_rates);
    popmatrix();
}

```

```

20 'abthesis
30 '11-3-86
40 DEFINT A-Z
45 ON KEY(1) GOSUB 999
46 KEY (1) ON
47 ON KEY(2) GOSUB 1000
50 DEF FNHIGHTRANS(STIC)=STIC*4
60 DEF FNLOWTRANS(STIC)=STIC*3
70 DEF FNPULSETRANS(STIC)=STIC
80 DEF FNROTATE(STIC)=STIC
90 DEF FNROTACCEL(STIC,RATE)=RATE+STIC
100 DEF FNROTPULSE(STIC,RATE)=RATE+STIC\ (ABS(STIC)+1)
101 CLS:WIDTH 80:KEY OFF
102 OPEN "com1:9600,e,7,1,cs,ds,cd" AS #1
103 LOCATE 3,5:PRINT "X Y Z"
104 LOCATE 5,3:PRINT X$
105 LOCATE 5,11:PRINT Y$
106 LOCATE 5,19:PRINT Z$
107 LOCATE 5,24:PRINT R$
108 LOCATE 5,31:PRINT P$
109 LOCATE 5,38:PRINT W$
110 'CHECK SWITCH STATES
120 OUT &H303,&H82
130 OUT &H300,&HFF
131 OUT &H302,&HF
132 OUT &H303,&H6
133 A=INP(&H301)
134 IF A=253 THEN PRINT #1,9 ELSE GOTO 131
140 OUT &H302,&HF
150 OUT &H303,&H6
160 A=INP(&H301)
170 OUT &H303,&H7
180 OUT &H303,&H4
190 B=INP(&H301)
200 OUT &H303,&H5
210 OUT &H303,&H2

```

Roll Pitch Yaw

'ports A and C output, B input
'port A pins high
'port C set for digital input
'set INT A* (active low)

'port C set for digital input
'set INT A* (active low)

'reset INT A* (active high)
'set INT B*

'reset INT B*
'set INT C*

```

220 C=INP(&H301)
230 OUT &H303,&H3
250 IF B=127 THEN X$="HIGH" ELSE IF B=191 THEN Y$="HIGH" ELSE IF B=223 THEN Z$="
HIGH" ELSE IF B=239 THEN R$="ACCEL" ELSE IF B=247 THEN P$="ACCEL" ELSE IF B=251
THEN W$="ACCEL"
260 IF C=127 THEN X$="PULSE" ELSE IF C=191 THEN Y$="PULSE" ELSE IF C=223 THEN Z$
="PULSE" ELSE IF C=239 THEN R$="PULSE" ELSE IF C=247 THEN P$="PULSE" ELSE IF C=2
51 THEN W$="PULSE"
270 IF A=127 THEN X$="LOW" ELSE IF A=191 THEN Y$="LOW" ELSE IF A=223 THEN Z$="LO
W" ELSE IF A=239 THEN R$="RATE" ELSE IF A=247 THEN P$="RATE" ELSE IF A=251 THEN
W$="RATE" ELSE IF A=253 THEN PRINT #1,99
300 'read hand controllers
310 'OUT &H303,&H82
320 'OUT &H300,&HFF
330
340 OUT &H302,&H80
350 OUT &H303,&H9
360 OUT &H303,&H8
370 OUT &H303,&HB
380 RSTICK=INP(&H301)
390 OUT &H302,&H81
400 OUT &H303,&H9
410 OUT &H303,&H8
420 OUT &H303,&HB
430 WSTICK=INP(&H301)
440 OUT &H302,&H82
450 OUT &H303,&H9
460 OUT &H303,&H8
470 OUT &H303,&HB
480 PSTICK=INP(&H301)
490 OUT &H302,&H83
500 OUT &H303,&H9
510 OUT &H303,&H8
520 OUT &H303,&HB
530 ZSTICK=INP(&H301)
540 OUT &H302,&H84

'reset INT C*
'ports A and C output, B input
'port A pins high

'analog line 0
'set PC4 and start convert
'reset PC4
'set PC5 and enable output

'analog line 1
'set PC4 and start convert
'reset PC4
'set PC5 and enable output

'analog line 2
'set PC4 and start convert
'reset PC4
'set PC5 and enable output

'analog line 3
'set PC4 and start convert
'reset PC4
'set PC5 and enable output

'analog line 4

```



```

550 OUT &H303,&H9
560 OUT &H303,&H8
570 OUT &H303,&HB
580 YSTICK=INP(&H301)
590 OUT &H302,&H85
600 OUT &H303,&H9
610 OUT &H303,&H8
620 OUT &H303,&HB
630 XSTICK=INP(&H301)
640 IF X$<> "PULSE" OR XCHECK=0 THEN IF XSTICK<100 THEN XINPUT=2 ELSE IF XSTICK>
154 THEN XINPUT=1 ELSE XINPUT=0
650 IF Y$<> "PULSE" OR YCHECK=0 THEN IF YSTICK<102 THEN YINPUT=2 ELSE IF YSTICK>
156 THEN YINPUT=1 ELSE YINPUT=0
660 IF Z$<> "PULSE" OR ZCHECK=0 THEN IF ZSTICK<103 THEN ZINPUT=2 ELSE IF ZSTICK>
157 THEN ZINPUT=1 ELSE ZINPUT=0
670 RINPUT=YSTICK\36-3
680 PINPUT=(PSTICK-8)\36-3
690 WINPUT=-(WSTICK-12)\36+3
700 IF X$="LOW" THEN STARBOARD=FNLOWTRANS(XINPUT) ELSE IF X$="HIGH" THEN STARBOA
RD=FNHIGHTTRANS(XINPUT) ELSE IF X$="PULSE" AND XCHECK=0 THEN STARBOARD=FNPUSETRA
NS(XINPUT) ELSE STARBOARD=0
710 IF Y$="LOW" THEN UPWARD=FNLOWTRANS(YINPUT) ELSE IF Y$="HIGH" THEN UPWARD=FNH
IGHTRANS(YINPUT) ELSE IF Y$="PULSE" AND YCHECK=0 THEN UPWARD=FNPUSETRANS(YINPUT
) ELSE UPWARD=0
720 IF Z$="LOW" THEN BACKWARD=FNLOWTRANS(ZINPUT) ELSE IF Z$="HIGH" THEN BACKWARD
=FNHIGHTTRANS(ZINPUT) ELSE IF Z$="PULSE" AND ZCHECK=0 THEN BACKWARD=FNPUSETRANS(
ZINPUT) ELSE BACKWARD=0
730 IF R$="RATE" THEN ROLLRATE=FNROTATE(RINPUT) ELSE IF R$="ACCEL" THEN ROLLRAT
E=FNROTACCEL(RINPUT,ROLLRATE) ELSE IF R$="PULSE" AND RCHECK=0 THEN ROLLRATE=FNRO
TPULSE(RINPUT,ROLLRATE) ELSE ROLLRATE=0
740 IF P$="RATE" THEN PITCHRATE=FNROTATE(PINPUT) ELSE IF P$="ACCEL" THEN PITCHR
ATE=FNROTACCEL(PINPUT,PITCHRATE) ELSE IF P$="PULSE" AND PCHECK=0 THEN PITCHRATE=
FNROTPULSE(PINPUT,PITCHRATE) ELSE PITCHRATE=0
750 IF W$="RATE" THEN YAWRATE=FNROTATE(WINPUT) ELSE IF W$="ACCEL" THEN YAWRATE=
FNROTACCEL(WINPUT,WRATE) ELSE IF W$="PULSE" AND WCHECK=0 THEN YAWRATE=FNROTPULSE
(WINPUT,YAWRATE) ELSE YAWRATE=0

```

```

771 PRINT #1, (STARBOARD*100)+(UPWARD*10)+BACKWARD
800 CLS
851 LOCATE 3,5:PRINT "X      Y      Z      Roll      Pitch      Yaw"
860 LOCATE 5,3:PRINT X$
880 LOCATE 5,11:PRINT Y$
890 LOCATE 5,19:PRINT Z$
891 LOCATE 5,31:PRINT R$
892 LOCATE 5,39:PRINT P$
893 LOCATE 5,47:PRINT W$
900 ON KEY(1) GOSUB 999
901 PRINT XINPUT;YINPUT;ZINPUT,,,RINPUT;PINPUT;WINPUT
902 PRINT STARBOARD;UPWARD;BACKWARD,,,ROLLRATE;PITCHRATE;YAWRATE
903 PRINT XCHECK;YCHECK;ZCHECK,,,RCHECK;PCHECK;WCHECK
904 PRINT (STARBOARD*100)+(UPWARD*10)+BACKWARD
905 'reset values
910 IF X$<> "PULSE" THEN XSTICK=127 ELSE IF XSTICK<154 AND XSTICK>100 THEN XCHECK
K=0 ELSE XCHECK=9
920 IF Y$<> "PULSE" THEN YSTICK=127 ELSE IF YSTICK<154 AND YSTICK>100 THEN YCHECK
K=0 ELSE YCHECK=9
930 IF Z$<> "PULSE" THEN ZSTICK=127 ELSE IF ZSTICK<154 AND ZSTICK>100 THEN ZCHECK
K=0 ELSE ZCHECK=9
940 IF R$<> "PULSE" THEN RSTICK=127 ELSE IF RSTICK<154 AND RSTICK>100 THEN RCHECK
K=0 ELSE RCHECK=9
950 IF P$<> "PULSE" THEN PSTICK=127 ELSE IF PSTICK<154 AND PSTICK>100 THEN PCHECK
K=0 ELSE PCHECK=9
960 IF W$<> "PULSE" THEN WSTICK=127 ELSE IF WSTICK<154 AND WSTICK>100 THEN WCHECK
K=0 ELSE WCHECK=9
970 GOTO 140
999 KEY (1) ON:KEY (2) ON:GOTO 999'end trial, flush buffer
1000 PRINT #1,9:RETURN 103'next trial

```

Appendix A.2--Transformation Matrices

$$\begin{bmatrix} \cot \left[\frac{\text{fovy}}{2} \right] & 0 & 0 & 0 \\ \frac{\text{aspect}}{\cot \left[\frac{\text{fovy}}{2} \right]} & 0 & 0 & 0 \\ 0 & 0 & -\frac{\text{far} + \text{near}}{\text{far} - \text{near}} & -1 \\ 0 & 0 & -\frac{2 \cdot \text{far} \cdot \text{near}}{\text{far} - \text{near}} & 0 \end{bmatrix}$$

Perspective

fovy = field of view in the y direction

aspect = y/x

far = distance to the far clipping plane

near = distance to the near clipping plane

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Roll} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Pitch} \quad \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Yaw}$$

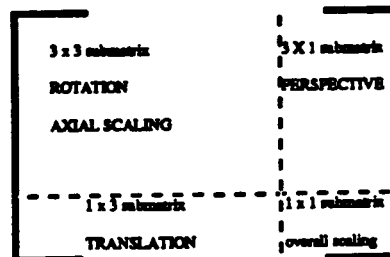
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

Translation

T_x = Translation along x-axis

T_y = Translation along y-axis

T_z = Translation along z-axis



Complete Matrix

(From IRIS User's Guide, Silicon Graphics, Inc., 1985)

Appendix A.3--NASA Approach Profile

<u>RANGE</u>		<u>CLOSURE RATE</u>	<u>TIME</u>
<u>INTERVAL</u>			
(meters)	(feet)	(ft/s)	(s)
304.8	1000-400	1.0	600
121.9	400-300	0.4	250
91.4	300-200	0.3	333
61.0	200-100	0.2	500
30.4	100-0	0.1	1000
			<hr/>
			2683

[Adapted from NASA/JSC RNDZ 2102, Rendezvous/Proximity Operations Workbook, 1983]

If performed exactly, this approach would cost \$13,522 in 1984\$ (for one astronaut) not including fuel costs. Since two people would be involved, the cost would be \$27,044. The addition of fuel costs would bring the total cost to about \$37,000.

Appendix A.4--Fuel Consumption Calculation

The governing equations for determining the fuel consumption levels are:

$$c = I_{sp} \cdot g_0$$

$$m = e^{-\Delta v/c} \cdot m_0$$

where c is the exhaust velocity of the rocket, I_{sp} is the specific impulse of the fuel, g_0 is the acceleration due to gravity at sea level (9.81 m/s^2), and m_0 is the initial vehicle mass. The fuel for the PRCS has an I_{sp} of 280.0 s which yields an exhaust velocity of 2749 m/s. The corresponding values for the OMS fuel are 313.2 s and 3071 m/s respectively. For a vehicle with an initial mass of 5000 kg, the PRCS uses 1.82 kg to produce a Δv of 1m/s (\$6370/m/s). The value for the OMS engines is 1.63 kg/m/s (\$5705/m/s).

The existence of a payload on the return trip will raise m_0 to about 10,000 kg. This doubling of vehicle mass requires a doubling of expelled propellant mass (thrust) to achieve the same Δv . Vehicles returning with satellites, or in general, heavier spacecraft, will probably travel more slowly since a greater deceleration force (additional fuel) would be required to null their velocities.

Appendix A.5--Test Subject Training Manual

Introduction

This experiment is designed to determine how initial (from 1000 ft) docking velocity affects docking performance. An emergency situation may arise where the pilot or vehicle or both must return safely as soon as possible. In addition, if it becomes possible to produce fuel inexpensively on orbit, the fuel multiplier in the cost function can be reduced drastically in which case a least time solution would be a least cost solution as well. Both reasons justify the search for a lowest time, safe docking approach. By having test subjects dock with an assortment of initial velocities, the effect of speed will be revealed. Performance measurements at mission completion are y offset, y rate error, and z rate error. Mission time, cost, and fuel consumption are also recorded.

Each subject must be committed to participating in six sessions. Each session should last approximately one-half hour.

Background

The vehicle which is being simulated is a small, one person spacecraft which can be used for retrieving satellites. An overhead view appears as Figure 1. The Digital Auto Pilot (DAP) (Figure 2) is used for identifying the current control mode for each Degree Of Freedom (DOF). For simplicity, only forward (tangential) and upward (radial) motions will be used. Forward (and backward) corresponds to "Z" while upward (and downward) are designated by "Y". To initiate a thrust, the left hand (translational) hand controller should be moved in the appropriate direction.

It costs \$3500 (1984\$) to get a kilogram of materials to the space station and each astronaut-day is valued at \$439,000. Since the final approach mandates a crewperson at the station monitoring the approach in addition to the pilot, each second of time is valued at \$10.16. The cost of fuel amounts to about \$6000/m/s.

Since there are no resistive forces operating on the spacecraft, the vehicle will travel at a constant velocity until reverse thrust is applied (or until significant orbital mechanics effects are encountered). Fuel is used most efficiently if the vehicle maintains its fastest speed for the longest amount of time. The integral of velocity with respect to time is distance travelled and since this is a constant, time is minimized by maintaining a high velocity. (See Figures 3a, 3b)

A successful docking is achieved by being in the right place at the right velocity. In the forward direction, the spacecraft must arrive between 1.5 and 2.0 meters from the target at a rate of 0.05-0.15 m/s. In the other directions, the misalignment can be no greater than 0.23 meters and the relative velocity must be less than 0.06 m/s. The range and range rate displays are color-coded to reveal to the pilot when the appropriate value has been reached. If the value is too low (or too slow) it appears in GREEN, if it is too large (or too fast) it appears in YELLOW, and just right is indicated by a RED display. While it is acknowledged that red usually connotes "hazard" rather than "condition met", the analogy to traffic signals is hoped to be a dominating intuitive influence for the test population.

A cyan docking target in the form of a "+" implemented with crosshairs appears at the station's center of mass. A 70% scale model in black is located 1 m closer to the approaching craft. When the spacecraft is correctly targeted, only the extremities of the cyan target are visible.

(See Figures 4a, 4b) This is a great aid in aiming the chaser vehicle. When the target fills the screen, the docking fixtures are correctly aligned.

A laser docking system located at the station's docking port also facilitates docking. This feature provides accurate range, range rate, angular position, and attitude of the chaser vehicle and these values are displayed near the bottom of the main screen.

The mission starts at a position of zero relative motion with respect to the space station 304.8 m in front of the station. From this position, orbital mechanics effects reduce the altitude of the chaser vehicle at a rate proportional to the forward velocity. Upward thrusts must be made to compensate for this.

At the completion of a successful docking mission, all motion stops and all controls are frozen. Pilot performance data are then displayed for feedback.

Strategy

Pressing the DAP ON button will begin the mission at the appropriate velocity. Control modes must then be entered for the Y and Z degrees of freedom. Good starting modes are HIGH for Z and LOW for Y, except for a starting velocity of 0.3 m/s, in which case LOW and PULSE respectively might prove more advantageous. These should progress to LOW and then PULSE for Z and to PULSE for Y as the range is reduced. It is advisable to maintain the Y values in a "condition met" (RED) situation throughout the mission to avoid excessive thrusting at mission end. It should be remembered that by maintaining the highest velocity for the majority of the mission, time will be minimized. Overthrusting should be avoided.

HIGH mode provides a Δv of 1.0 m/s, the LOW value is 0.1 and PULSE produces 0.05 m/s.

Missions will begin with initial velocities ranging from 0.3 to 9.0 m/s. Two "safe" (non crash) runs will be performed at 0.3 m/s and four sessions of ten runs each will begin at 3.0, 5.0, 7.0, and 9.0 m/s respectively. One session without data displays will be performed at 3.0 m/s. Each session should last approximately 30 minutes.

Make sure a "10 % rule" is not exceeded in the final 100 m. For example, make sure the forward velocity is below 3 m/s at a range of 30 m.

The interested test subject is referred to the test conductor's thesis for greater detail and additional information.

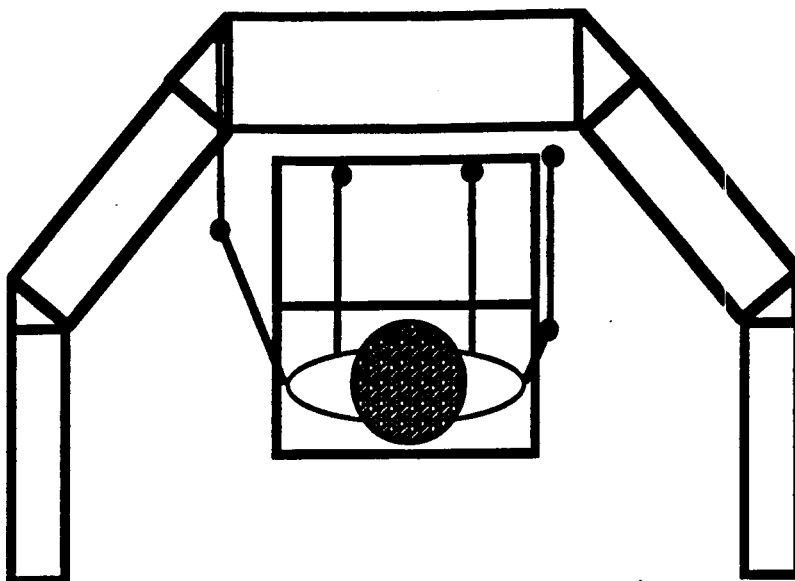


Figure A.5.1: Cockpit (top view)

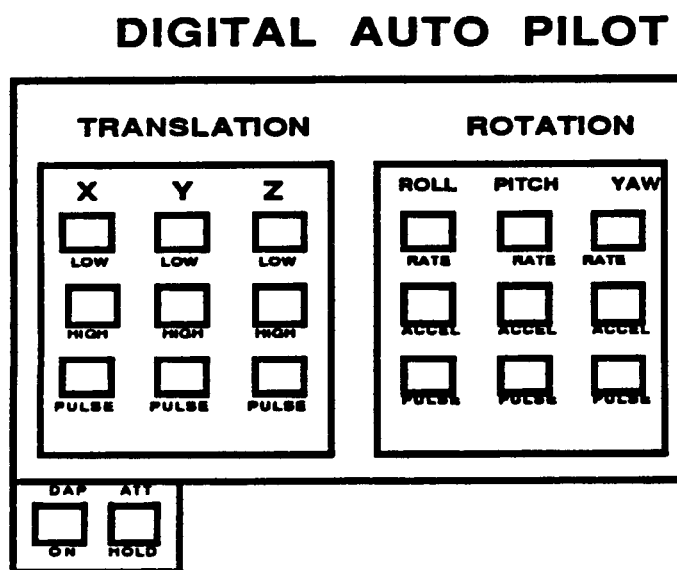
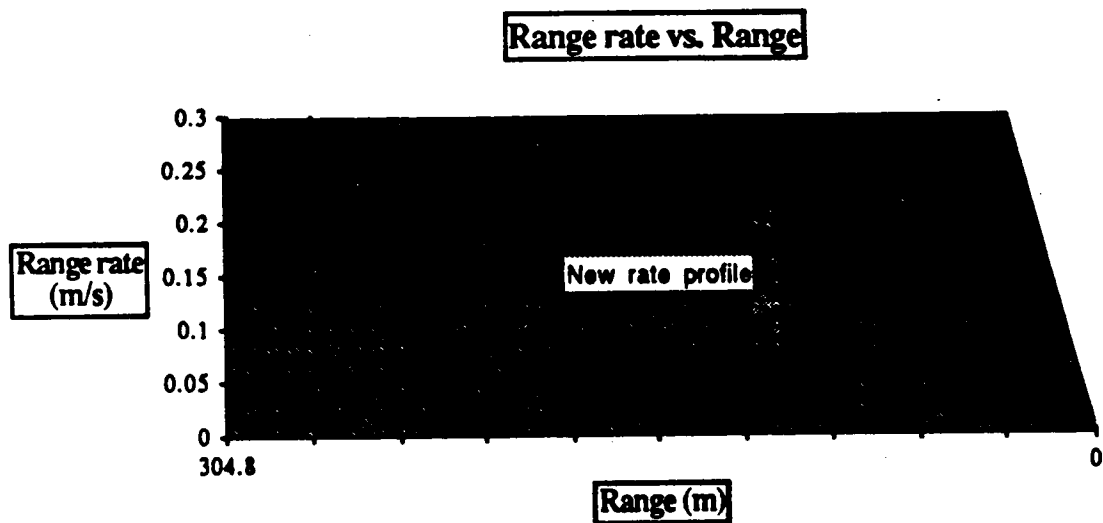
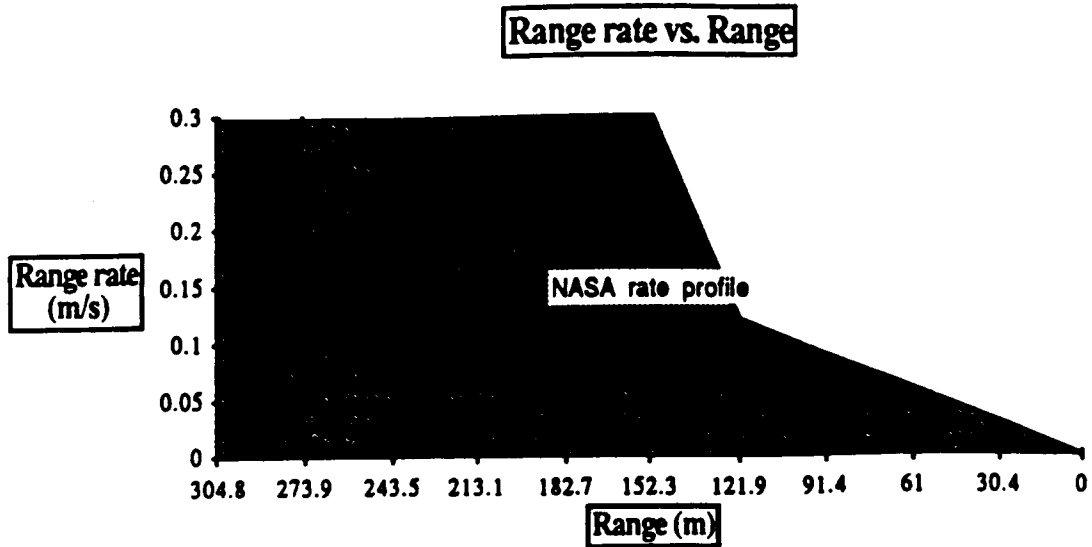


Figure A.5.2: Digital Auto Pilot



Note that these illustrate the rate profile in terms of range and not time. Since the range is constant for both approaches, performance is optimized by maximizing the area under the curve.

Figures A.5.3a, A.5.3b: Range Rate vs. Range

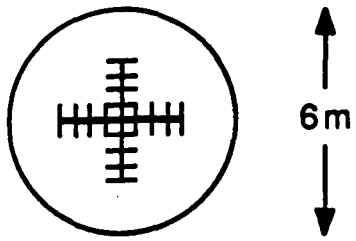


Figure A.5.4a: Docking Target

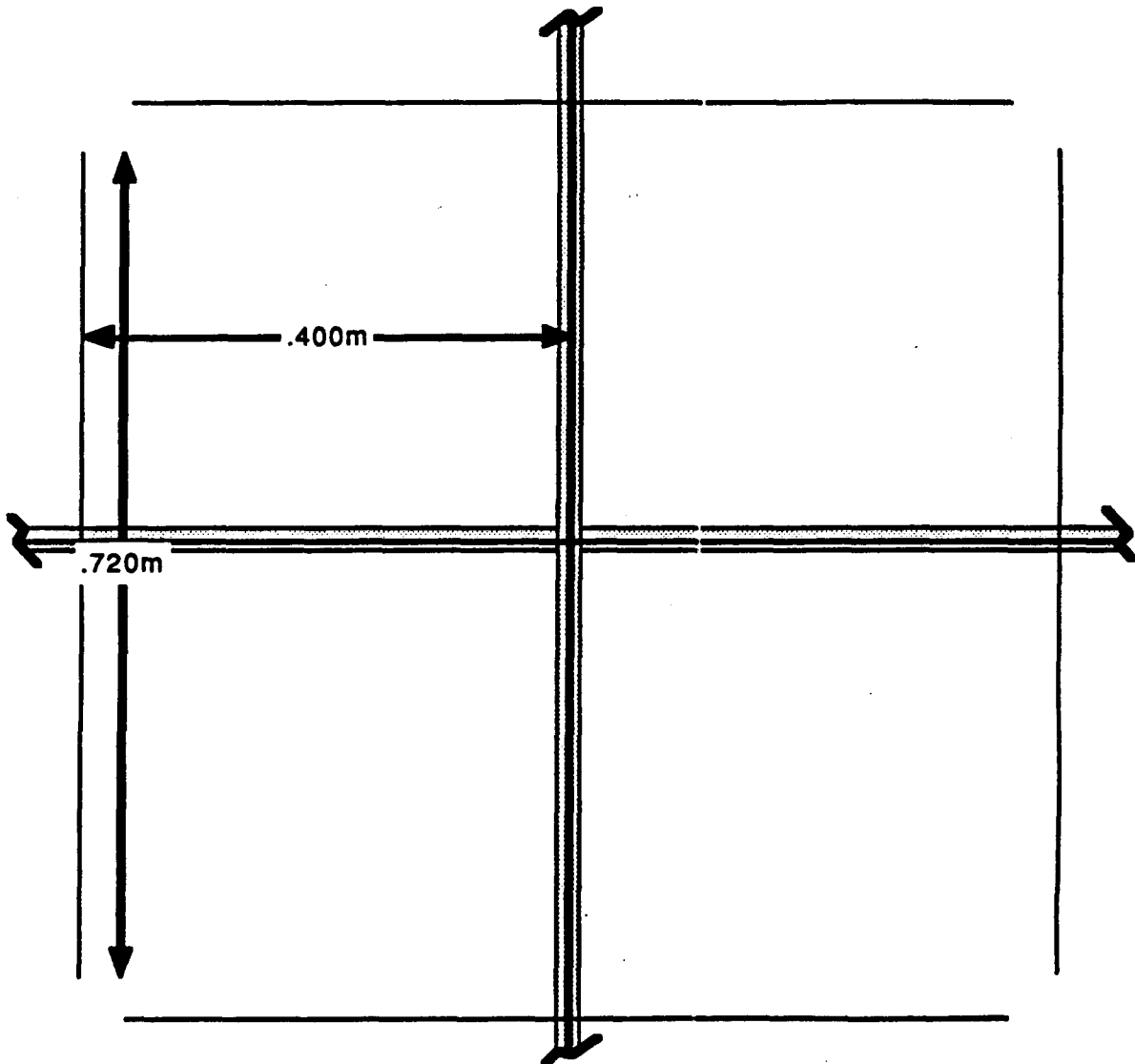


Figure A.5.4b: Docking Target (detail)

Appendix A.6--Raw Data

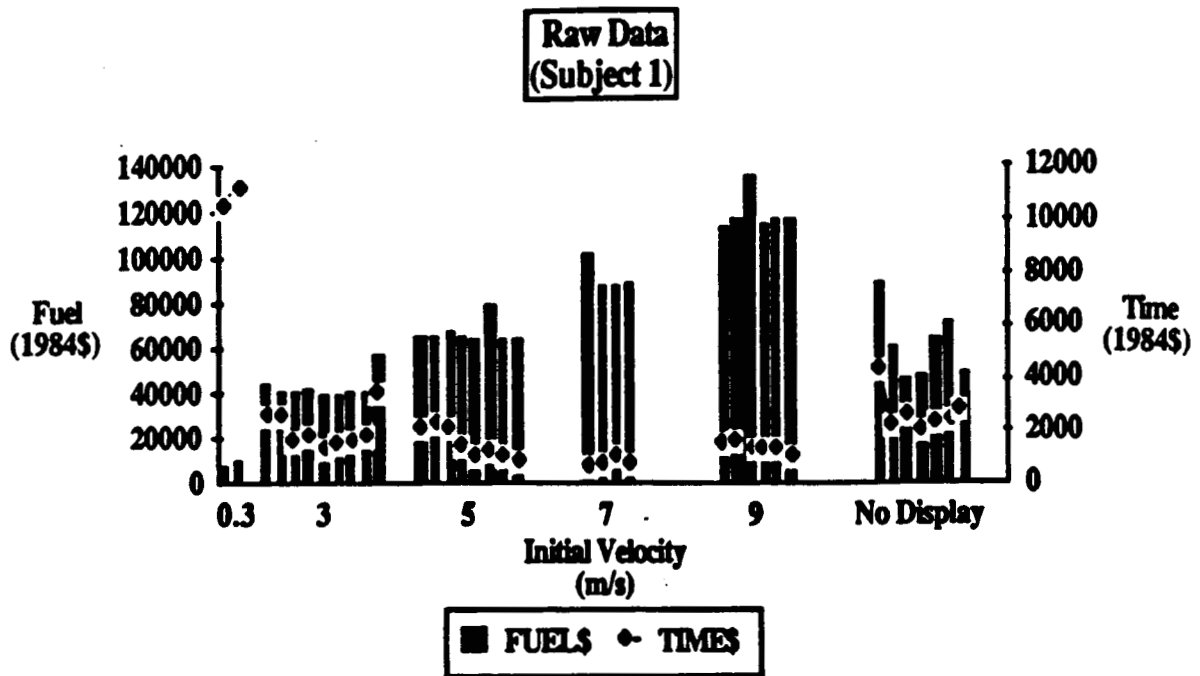


Figure A.6.1: Raw Data--Test Subject 1

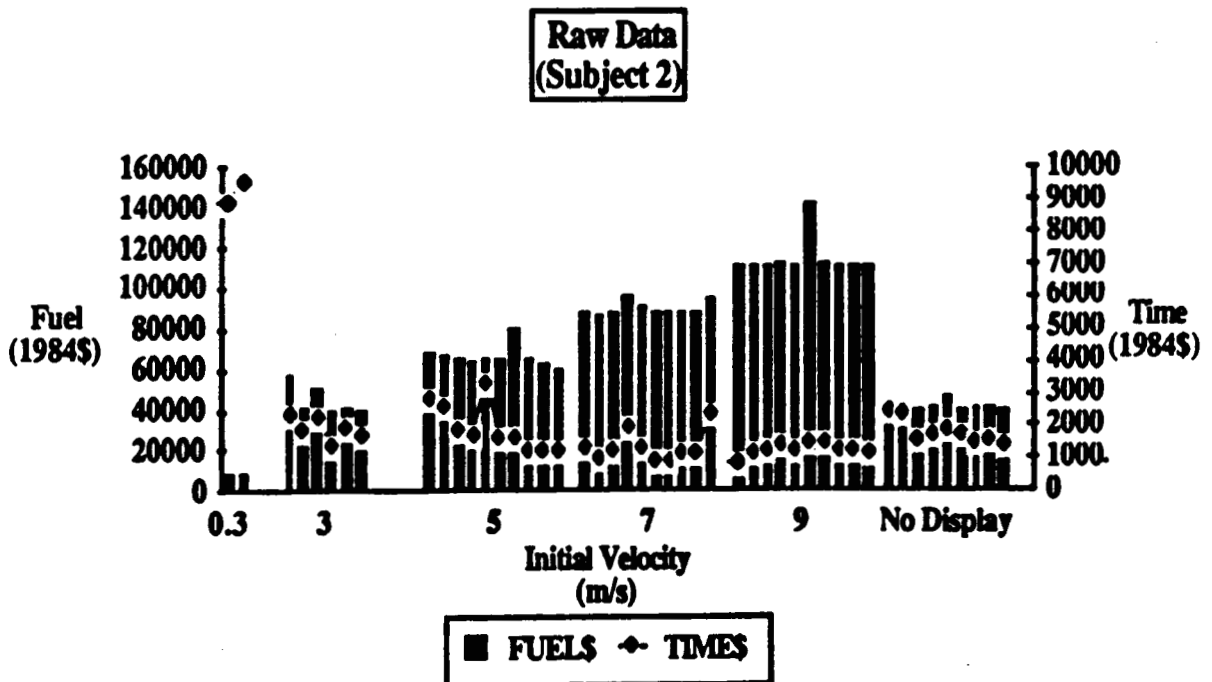


Figure A.6.2: Raw Data--Test Subject 2

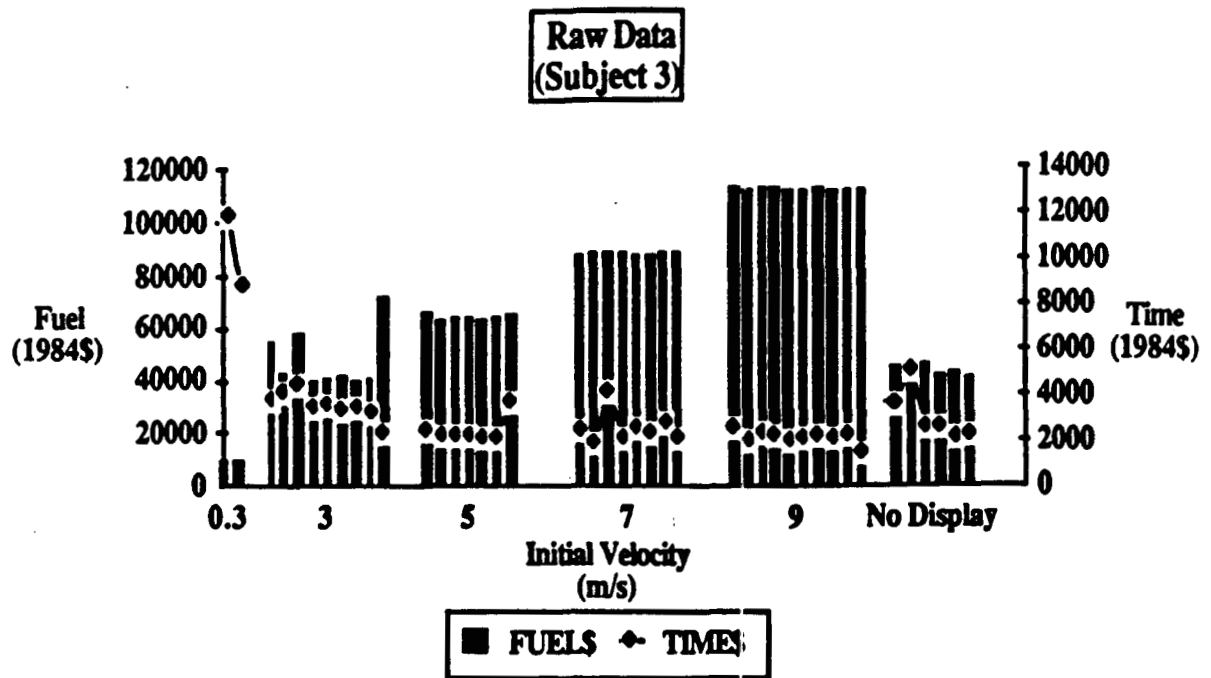


Figure A.6.3: Raw Data--Test Subject 3

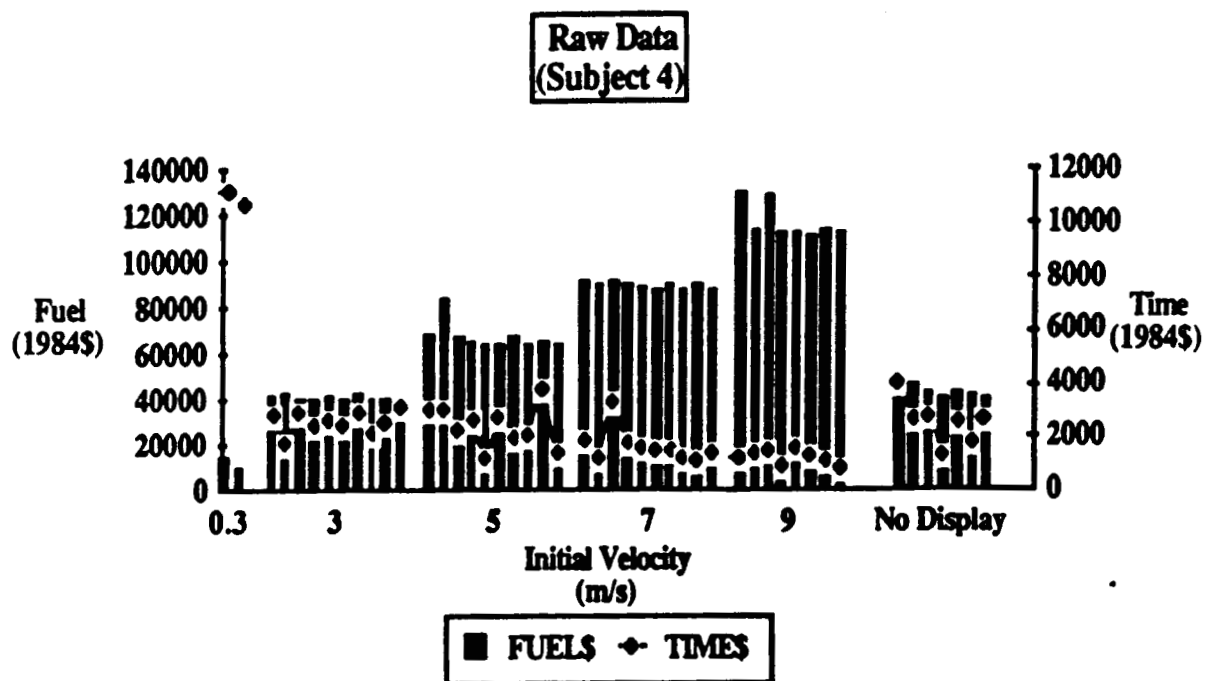


Figure A.6.4: Raw Data--Test Subject 4

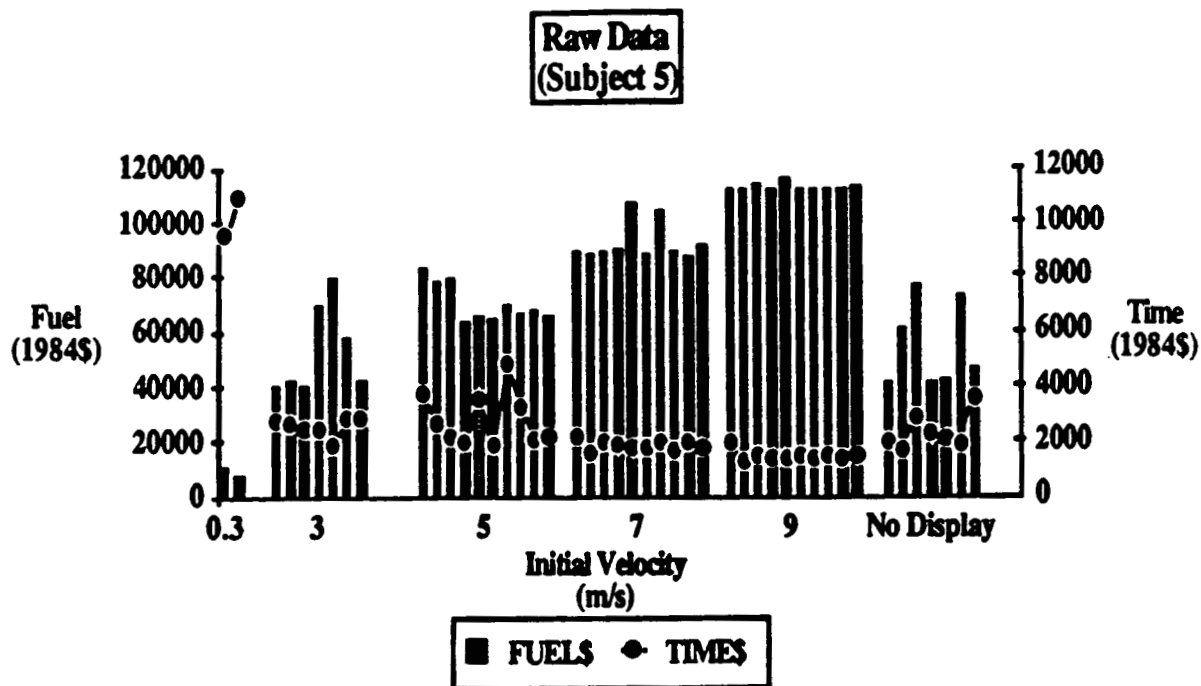


Figure A.6.5: Raw Data--Test Subject 5

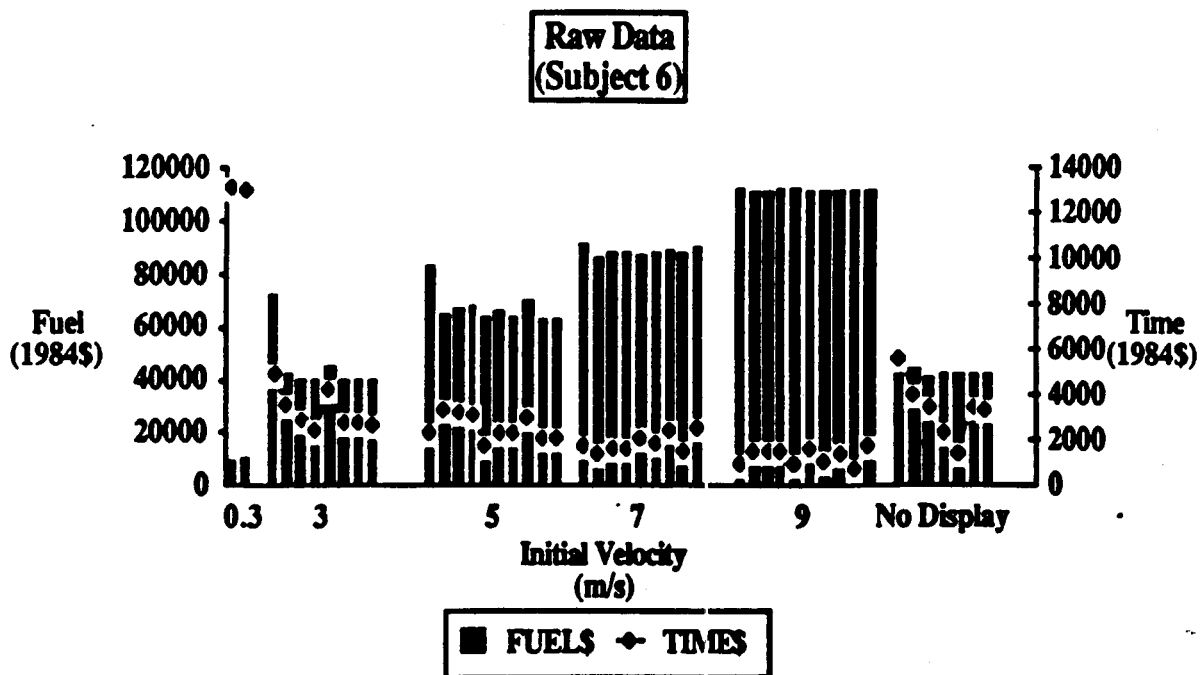


Figure A.6.6: Raw Data--Test Subject 6

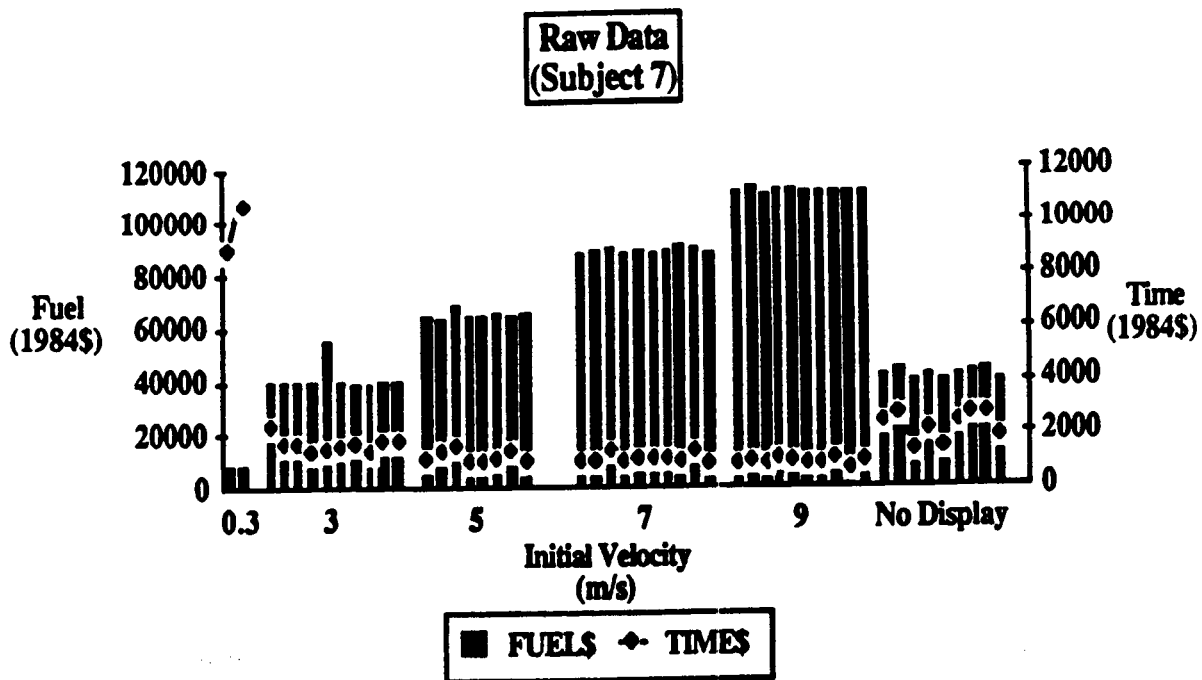


Figure A.6.7: Raw Data--Test Subject 7

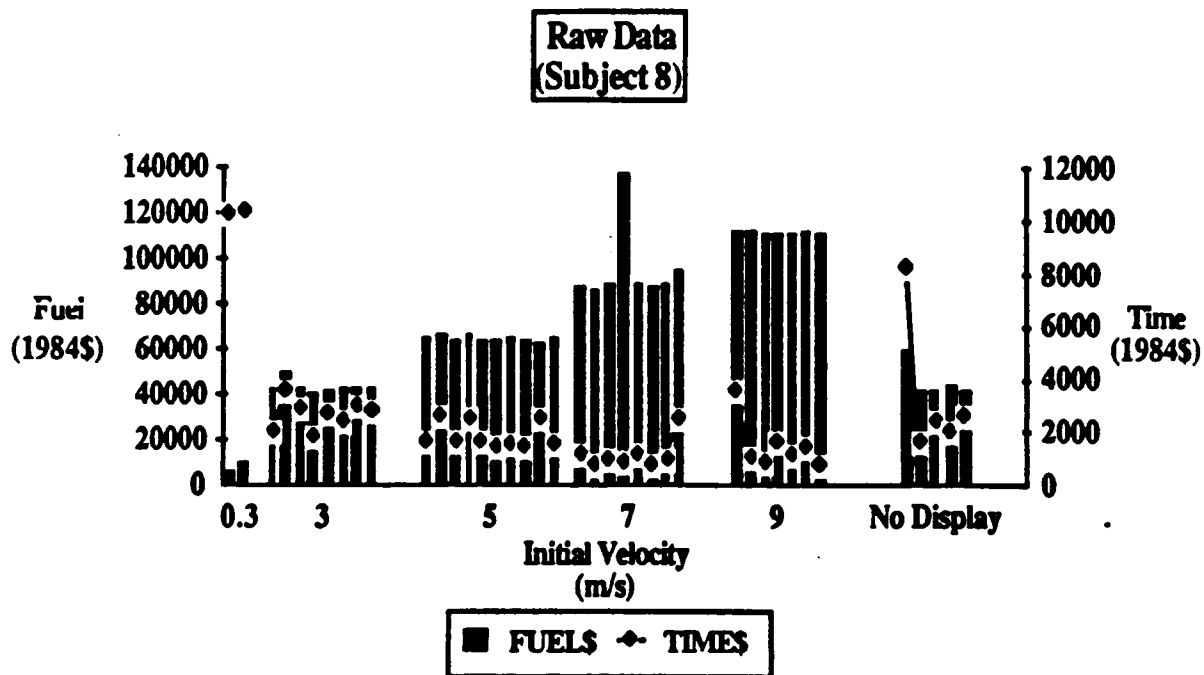


Figure A.6.8: Raw Data--Test Subject 8

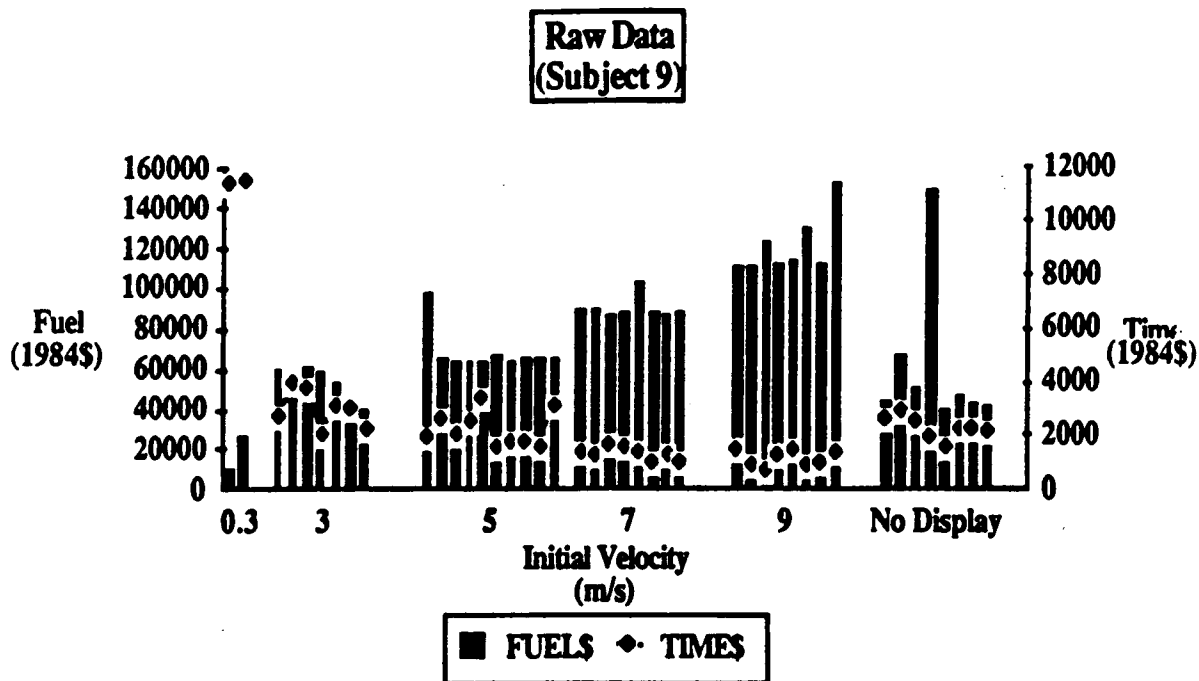


Figure A.6.9: Raw Data--Test Subject 9

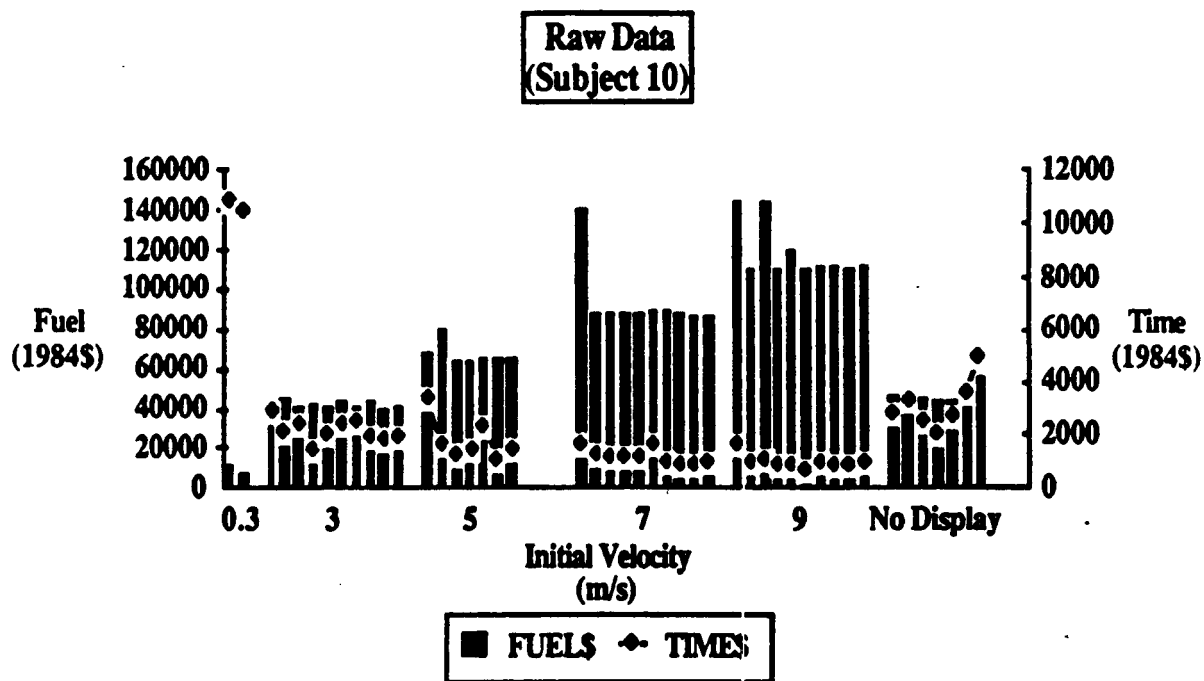


Figure A.6.10: Raw Data--Test Subject 10

References Cited

- Hughes, Frank and Charles Floyd. "Navigation Training for Current Space Flights", *Navigation*, Vol. 17, No. 1, Spring 1970.
- Kaplan, Marshall H. Modern Spacecraft Dynamics & Control. New York: John Wiley & Sons, 1976.
- Larson, Vernon R. and Stephen A. Evans. "Propulsion for the Space Station", IAF-86-182, October, 1986.
- McCormick, Ernest J. and Mark Sanders. Human Factors in Engineering and Design. New York: McGraw Hill Book Co., 1982.
- NASA Lyndon B. Johnson Space Center. Proceedings of the Rendezvous and Proximity Operations Workshop, 1985.
- NASA Lyndon B. Johnson Space Center. Rendezvous/Proximity Operations Workbook, RNDZ 2102, 1983.
- NASA Lyndon B. Johnson Space Center *Systems Engineering and Integration Space Station Program Office*. Space Station Reference Configuration Description, 1984.
- NASA George C. Marshall Space Flight Center. "Analysis and Selection of a Remote Docking Simulation Visual Display", NASA-CR-171148, Aug. 1984.
- NASA Office of Technology Utilization *Scientific and Technical Information Division*. "Gemini Summary Conference", NASA SP-138, Feb. 1967.
- Novikov, Yu and B. Fedorov. "Docking in Space--A Complex Problem", *Aviatsiya i Kosmonavtika*, February 1968.
- Panero, Julius and Martin Zelnick. Human Dimension and Interior Space. New York: Whitney Library of Design, 1979.
- Sekuler, Robert and Randolph Blake. Perception. New York: Alfred A. Knopf, 1985.
- Stuart, David G. A Systems Analysis of Humans and Machines in Space Activities, MIT Sc.D. Thesis, June 1986.



Report Documentation Page

1. Report No. NASA CR 177502		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Spacecraft Flight Simulation: A Human Factors Investigation into the Man-Machine Interface between an Astronaut and a Spacecraft Performing Docking Maneuvers and Other Proximity Operations				5. Report Date September 1988	
				6. Performing Organization Code	
7. Author(s) Adam R. Brody				8. Performing Organization Report No.	
				10. Work Unit No. 506-47-31	
9. Performing Organization Name and Address Sterling Federal Systems, Inc. <i>Massachusetts Institute of Technology, Space Systems Lab</i> 1121 San Antonio Road <i>Cambridge, MA 02139</i> Palo Alto, CA 94303-4380				11. Contract or Grant No. NAS2-11555 <i>NAGW-21</i>	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				13. Type of Report and Period Covered Contractor Report	
				14. Sponsoring Agency Code	
15. Supplementary Notes Point of Contact: Robert Carlson MS233-15 Ames Research Center Moffett Field, CA 94035 FTS 448-6036 (415) 694-6036					
16. Abstract The anticipated increase in rendezvous and docking activities in the various space programs in the Space Station era necessitates a renewed interest in manual docking procedures. Ten test subjects participated in computer simulated docking missions in which the influence of initial velocity was examined. All missions started from a resting position of 304.8 meters (1000 feet) along the space station's +V-bar axis. Test subjects controlled their vehicle with a translational hand controller and digital auto pilot which are both virtually identical to their space shuttle counterparts. While the "0.1% rule" (range rate is equal to 0.1% of the range) used by space shuttle pilots is comfortably safe, it is revealed to be extremely inefficient in terms of time and not justifiable in terms of marginal safety. Time is worth money, not only because of training and launch costs, but because the sooner a pilot and spacecraft return from a mission, the sooner they can begin the next one. Inexperienced test subjects reduced the costs of simulated docking by close to a factor of 2 and achieved safe dockings in less than 4% of the time the baseline approach would entail. This reduction in time can be used to save lives in the event of an accident on orbit, and can tremendously reduce docking costs if fuel is produced from waste water on orbit. Test subjects had little additional difficulty performing successful missions when numerical range and range rate displays were unavailable, demonstrating that, while improving performance and reducing fatigue, such displays are not essential for a successful docking. For a safe docking, therefore, it is not necessary to equip every potential docking target with an expensive laser docking device to provide these data. Experienced NASA pilots should have no difficulty exceeding these achievements and should be encouraged to perform similar simulated dockings to further evaluate the feasibility of new profiles.					
17. Key Words (Suggested by Author(s)) Simulation, Docking, PROX OPS			18. Distribution Statement Unclassified, unlimited Star Category - 53		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 96	22. Price A05